

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

**Avaliação da experiência do usuário com base na
arbitragem de parâmetros de QoS em fluxo Multimídia, em
cenários com roteadores Cisco e Placa NetFPGA**

Rafael Sant’Ana Jacaúna

SÃO CRISTÓVÃO/ SE

2016

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

Rafael Sant’Ana Jacaúna

Avaliação da experiência do usuário com base na
arbitragem de parâmetros de QoS em fluxo Multimídia, em
cenários com roteadores Cisco e Placa NetFPGA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Edward David Moreno Ordonez

Co-Orientador: Prof. Dr. Ricardo José P. de B. Salgueiro

SÃO CRISTÓVÃO/ SE

2016

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

- J12a Jacaúna, Rafael Sant'Ana
Avaliação da experiência do usuário com base na arbitragem de parâmetros de QoS em fluxo multimídia, em cenários com roteadores Cisco e placa NetFPGA / Rafael Sant'Ana Jacaúna ; orientador Edward David Moreno Ordoñez. - São Cristóvão, 2016. 103 f. : il.
- Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Sergipe, 2016.
1. Tecnologia streaming (Telecomunicações). 2. Roteadores (Rede de computadores). 3. Redes de computadores - Protocolos. 4. TCP/IP (Protocolo de rede de computadores). 5. Fluxo de dados (Computadores). I. Ordoñez, Edward David Moreno, orient. II. Título.

CDU 004.715.623

Rafael Sant'Ana Jacaúna

**Avaliação da experiência do usuário com base na
arbitragem de parâmetros de QoS em fluxo Multimídia, em
cenários com roteadores Cisco e Placa NetFPGA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS) como parte de requisito para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Edward David Moreno Ordonez, Presidente
Universidade Federal de Sergipe (UFS)

Prof. Dr. Ricardo José P. de B. Salgueiro, Co-Orientador
Universidade Federal de Sergipe (UFS)

Prof. Dr. Wanderson Roger Azevedo Dias, Membro
Instituto Federal de Sergipe (IFS)

Prof. Dr. Fabio Dacencio Pereira, Membro
Centro Universitário Eurípides de Marília (UNIVEM)

Agradecimentos

À Deus, por tudo de bom que acontece (sempre presente) em minha vida. As experiências vividas, através de provas e expiações, fazem com que eu fique um pouco mais humano, sem deixar de lado a razão, o melhor juiz para a minha consciência.

Aos meus amigos orientadores, Edward e Salgueiro, cada um com seu estilo de ver e pensar sobre a vida e a pesquisa, souberam, com muito tato e sutileza, apoiar as minhas iniciativas, mas colocando-me no lugar certo quando eu começava sair da direção necessária para alcançar o objetivo final: tornar-me um ser humano melhor do que fui quando cheguei ao programa e, a partir da obtenção do diploma, que ainda há muito o que aprender. Sem querer desmerecer a todos os docentes do colegiado, afirmo sem a menor dúvida que vocês são a minha melhor experiência na carreira que resolvi abraçar como docente. Obrigado, queridos amigos!

A minha esposa, que como eu é docente, só que da área de humanas, soube incentivar-me da forma certa em todos os momentos difíceis que passei, as vezes de muita descrença naquilo que estava fazendo, mas sempre mostrando a outra face do esforço já cumprido. E quantas horas ouvia, pacientemente, sobre siglas e mais siglas de protocolos! Mas, sempre com bom humor, procurava tirar o peso das minhas costas com sua alegria peculiar. Meu anjo de guarda, amo você!

João e Bia, o mestrado finalmente acabou! Mas, ainda há outra etapa que desejo cumprir. Até lá, vamos fazer um passeio mais próximo, ok? (risos)

Aos meus filhos, que mesmo não convivendo diretamente comigo, entenderam algumas ausências inerentes dos estudos. Estamos no caminho certo, o tempo é o senhor da razão. Amo vocês, e amar é dar liberdade a quem você ama!

Aos meus colegas de trabalho, que compreenderam os meus períodos de ausências e ajudaram-me a cumprir com algumas atribuições na coordenação, em particular a Ana Carla, sempre apoiando-me quando precisei ausentar-me. E aos colegas professores de Informática, muito solidários em todos os momentos, em especial Alan, que foi a pessoa que mais ouviu as minhas "teorias" das coisas, com contribuições significativas. Obrigado a todos!

Aos meus amigos do BANESE, que sempre deram suporte quando as dúvidas surgiam sobre determinadas configurações das ferramentas de monitoramento de rede, e que, ao doarem os roteadores da Cisco para as aulas práticas do IFS, ajudou-me na realização dos experimentos dessa dissertação. O meu muito obrigado a vocês!

A minha "irmãinha" Celeste e meu compadre e cunhado Robério, não terei palavras para agradecer tanto apoio recebido.

Aos alunos da graduação em Computação da UFS, Itauan e Florêncio, que, de forma tímida, mas muito eficaz, ajudaram-me a corrigir alguns erros durante o processo de instalação do *software* da placa NetFPGA. O meu muito obrigado a vocês dois!

A todos os demais que, direta ou indiretamente, contribuíram nessa jornada de desafios, mas de muitos aprendizados também.

Resumo

Um *streaming* de vídeo tem características que o difere dos demais fluxos: a ocupação de grande quantidade de banda, e a possível variação de cenas com consequente aumento (ou diminuição) na quantidade dos *bytes* transmitidos. Aplicações para transmissão de *streaming* de vídeo (YouTube, Vimeo, NetFlix, Telecine Play, e etc.) tem despertado o interesse da comunidade científica quanto ao comportamento das redes. Nossa proposta é correlacionar, a partir de um fluxo de vídeo, a experiência do usuário com base na arbitragem de parâmetros de QoS em ambiente controlado. O resultado dos experimentos demonstrou a eficácia do protocolo TCP sobre o UDP, que na fase da escolha do protocolo mais adequado para os testes, mostrou-se ineficiente até a taxa *Gigabit Ethernet*. Esses experimentos ocorreram em topologias físicas distintas; apenas com roteadores Cisco, com e sem QoS, tendo ou não injeção de tráfego gerado pelo *software* *iPerf* na rede, e usando a placa NetFPGA no modo *router*, apenas com e sem tráfego via *iPerf*, uma vez que o projeto desenvolvido para fazer a placa atuar como roteador não previu configurações de QoS. Nesses cenários, constatou-se a necessidade de iniciar a execução dos clientes em momentos diferentes, configurando seus respectivos *caches* de redes diferentes, para que não coincidisse as cenas, evitando altos picos de banda em trechos com muita movimentação.

Palavras-chave: *Streaming*. QoS. QoE. TCP. UDP. NetFPGA. *iPerf*.

Abstract

A streaming video has features that differs from other streams: the occupation of large amount of bandwidth, and the possible variation of scenes with consequent increase (or decrease) in amount of bytes transmitted. Applications for video streaming transmission (YouTube, Vimeo, Netflix, Telecine Play, etc.) have aroused the interest of the scientific community regarding the behavior of networks. Our proposal is to measure, from a video stream, the user experience (QoE) based on arbitration QoS parameters in a controlled environment which use NetFPGA. The results of the experiments has shown the efficacy of TCP over UDP, which at the stage of choosing the most appropriate protocol for testing, showed to be ineffective to the Gigabit Ethernet rate. These experiments occurred in distinct physical topologies; only Cisco routers, with and without QoS, with and without injection of traffic generated by software `iperf` in the network, and using NetFPGA board set to router only with and without traffic via `iperf`, once the design developed to make the board work as a router does not allow QoS settings. in these scenarios, there was the need to start the client running at different times, setting their respective caches different networks so that did not coincide scenes, avoiding high bandwidth peaks in stretches with a lot of movement.

Keywords: Streaming. QoS. QoE. TCP. UDP. NetFPGA. iPerf.

Lista de Figuras

2.1	Modelo de enfileiramento de tráfego para 3 filas e 3 classes	10
2.2	Visão lógica de um classificador de pacotes e Condicionador de Tráfego . .	11
2.3	Representação do Encaminhamento Expedido	15
2.4	Modelo QoE considerando fatores do ambiente e equipamento	21
2.5	Modelo QoE baseado na arbitragem de recursos	23
3.1	Placa NetFPGA-1G-CML	27
3.2	Arquitetura Kintex XC7K325T-1FFG676	28
3.3	Características de energização da placa	29
3.4	Diagrama de funcionamento do projeto NIC	33
4.1	Topologia usando a placa NetFPGA e gerador de tráfego iPerf	40
4.2	Quadro representando os conjuntos de filas nos roteadores Cisco	42
4.3	Fluxo do pacote marcado com prioridade	43
4.4	Estatísticas do <i>software</i> VLC. Cliente.	46
4.5	Estatísticas do <i>software</i> VLC a 1 Gbps	51
5.1	Modelo de QoE considerando equipamentos e ambiente como fatores de in- terferência.	57
5.2	Camada de Arbitragem.	58
5.3	Foto. Servidor VLC e PC com placa NetFPGA.	60
5.4	Roteadores Cisco e dispositivo multipropósito TP-LINK.	60
5.5	Interfaces de rede da placa NetFPGA	61
5.6	Interfaces de rede da placa NetFPGA	61
5.7	Sentido dos fluxos para aplicação de QoS nos roteadores Cisco	62

5.8	Estatísticas VLC do Ultrabook	65
5.9	Estatísticas VLC do HP Pro One 400 G1 AiO	65
5.10	Estatísticas Interface de Rede Servidor de <i>streaming</i>	66
5.11	Carga da interface de rede no servidor de <i>streaming</i>	68
5.12	Tráfego <i>iPerf</i> no roteador Cisco, lado cliente	69
5.13	Gráfico de transmissão a partir da interface de rede do servidor de <i>streaming</i>	74
5.14	Informações da interface de rede do Ultrabook	75
5.15	Carga na interface de rede do servidor de <i>streaming</i>	76
A.1	Formulário para avaliar a experiência do usuário.	87

Lista de Tabelas

2.1	Medidas Objetivas	18
2.2	Medidas Subjetivas	19
5.1	Características dos dispositivos utilizados no experimento	63
5.2	Carga gerada pelo iPerf e descartada pela política de QoS	70
5.3	Experiência do usuário – Protocolo HTTP	71
5.4	Experiência do dispositivo – Protocolo HTTP	71
5.5	Experiência do usuário – Protocolo RTSP	72
5.6	Experiência do dispositivo – Protocolo RTSP	72

Lista de Siglas

ACL	<i>Access Control List</i>
AF	<i>Assured Forwarding</i>
AQoS	<i>Application Quality of Service</i>
ARP	<i>Address Resolution Protocol</i>
BPI	<i>Byte Peripheral Interface</i>
CBWFQ	<i>Class Based Weighted Fair Queueing</i>
CCA	<i>Catalog, Categorize, and Analyze</i>
CDMA	<i>Code Division Multiple Access</i>
CLI	<i>Command Line Interface</i>
CPU	<i>Central Processor Unit</i>
CRC	<i>Cyclic Redundancy Check</i>
DDR3	<i>Double-Data Rate</i>
DiffServ	<i>Differentiated Service</i>
DMA	<i>Direct Memory Access</i>
DRAM	<i>Dynamic Random Access Memory</i>
DS	<i>Differentiated Service</i>
DSCP	<i>DiffServ Code Point</i>
DSFIELD	<i>DiffServ Field</i>
EF	<i>Expedited Forwarding</i>
E/S	<i>Entrada e Saída</i>
FIFO	<i>First In, First Out</i>
FPGA	<i>Field Programmable Gate Array</i>
FQ	<i>Fair Queueing</i>
FTP	<i>File Transfer Protocol</i>
Gbps	<i>Gigabit per second</i>

HD	<i>High Definition</i>
HDL	<i>Hardware Description Language</i>
http	<i>HyperText Transfer Protocol</i>
ICMP	<i>Internet Control Message Protocol</i>
IETF	<i>Internet Enginnering Taks Force</i>
IntServ	<i>Integrated Service</i>
IP	<i>Internet Protocol</i>
IPTV	<i>Internet Protocol TeleVision</i>
IQX	<i>(exponential interdependency of quality</i>
ISP	<i>Internet Service Provider</i>
ISR	<i>Integrated Services Router</i>
ITU	<i>International Telecommunication Union</i>
LAN	<i>Local Area Network</i>
LLQ	<i>Low Latency Queueing</i>
NetFPGA	<i>Network Field Programmable Gate Array</i>
MAC	<i>Media Access Control</i>
Mbps	<i>Millions of bits per second</i>
MIB	<i>Management Information Base</i>
MIG	<i>Management Interface Generator</i>
MOS	<i>Mean Opinion Score</i>
MPEG	<i>Moving Picture Experts Group</i>
MTU	<i>Maximun Transmission Unit</i>
NIC	<i>Network Interface Card</i>
NQoS	<i>Network Quality of Service</i>
NS-2	<i>Network Simulator version 2</i>
OSPF	<i>Open Shortest Path First</i>
OSI	<i>Open Systems Interconnect</i>
PCIe	<i>Peripheral Component Interconnect express</i>
PFS	<i>Proportional Fair Scheduling</i>
PHB	<i>Per-Hop Behaviors</i>
ppi	<i>pixels per inch</i>
QBS	<i>QoE Based Scheduling</i>

QDRI I	<i>Quad Data Rate</i>
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
RED	<i>Random Early Drop</i>
RFC	<i>Request for Comment</i>
RGMI I	<i>Reduced Gigabit Media Independent Interface</i>
RSVP	<i>Resource Reservation Protocol</i>
RTP	<i>Real-time Transport Protocol</i>
RTSP	<i>Real-time Transport Streaming Protocol</i>
RST	<i>Rough Set Theory</i>
SCONE	<i>Software Component Of NEtFPGA</i>
SD	<i>Secure Digital</i>
SDK	<i>Software Development Kit</i>
SDRAM	<i>Synchronous Dynamic Random Access Memory</i>
SNMP	<i>Simple Network Management Protocol</i>
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random Access Memory</i>
SSD	<i>Solid State Drive</i>
TCP	<i>Transport Control Protocol</i>
ToS	<i>Type of Service</i>
TTL	<i>Time to Live</i>
UDP	<i>User Datagram Protocol</i>
UE	<i>User Equipment</i>
USB	<i>Universal Serial Bus</i>
VCD	<i>Video Compact Disc</i>
VLAN	<i>Virtual Local Area Network</i>
VLC	<i>VideoLAN Client</i>
VoIP	<i>Voice over Internet Protocol</i>
VQM	<i>Video Quality Metric)-based Mapping Model</i>
WAN	<i>Wide Area Network</i>
WFQ	<i>Weighted Fair Queueing</i>

Sumário

1	Introdução	1
1.1	Problemática e Hipótese	3
1.2	Objetivos da Dissertação	4
1.2.1	Objetivos Específicos	4
1.3	Material e Métodos	4
1.4	Contribuições da Dissertação	6
1.5	Organização da Dissertação	6
2	Identificação de serviços da camada de rede que afetam a experiência do usuário	7
2.1	Qualidade de Serviços – QoS	8
2.2	Técnicas de Enfileiramento	9
2.3	Qualidade de Experiência – QoE	18
2.4	Trabalhos Correlatos	18
2.5	Considerações finais do Capítulo	24
3	Integração entre Placa NetFPGA e Roteadores Cisco	26
3.1	O modelo NetFPGA-1G-CML	26
3.2	Arquitetura da placa	32
3.3	Roteadores Cisco modelo 1841	35
3.4	Integração entre NetFPGA e Cisco	36
3.5	Considerações finais do Capítulo	37
4	Ambiente de testes	38
4.1	Ambiente para uso da placa NetFPGA	39
4.1.1	A escolha do Hardware	40

4.1.2	A escolha do Sistema Operacional	41
4.2	Configuração das classes de serviços nos roteadores Cisco	41
4.2.1	A escolha do software para simulação de vídeo sob demanda	43
4.3	Caminho percorrido para os testes	44
4.3.1	Determinação da Linha de Base	45
4.4	Teste dos protocolos	48
4.5	Métricas da Qualidade de Experiência	53
4.5.1	Configuração dos dispositivos dos observadores e procedimentos de avaliação	53
4.6	Considerações finais do Capítulo	55
5	Avaliação das condições perceptuais do usuário em transmissão de fluxo de da- dos	56
5.1	Execução do experimento	58
5.1.1	Testes no IFS	59
5.1.2	Testes na UFS	63
5.2	Cenário com roteadores Cisco nas instalações do IFS	64
5.2.1	Sem QoS nas Interfaces de rede dos roteadores, sem carga do iPerf	64
5.2.2	Sem QoS nas Interfaces de rede dos roteadores, com carga do iPerf	66
5.2.3	Com QoS nas Interfaces de rede dos roteadores, sem carga do iPerf	67
5.2.4	Com QoS nas Interfaces de rede dos roteadores, com carga do iPerf	67
5.3	Cenário com roteadores Cisco nas instalações da UFS	68
5.3.1	Com QoS nas Interfaces de rede dos roteadores, com e sem carga do iPerf	69
5.4	Cenário com a placa NetFPGA	73
5.4.1	Sem QoS nas Interfaces da placa NetFPGA, sem carga do iPerf .	73
5.4.2	Sem QoS nas Interfaces da placa NetFPGA, com carga do iPerf .	73
5.5	Considerações finais do Capítulo	74
6	Considerações Finais	77
6.1	Trabalhos Futuros	78

Capítulo 1

Introdução

Na Internet, há diversos fluxos de informações percorrendo rotas, com enlaces a taxas de transmissão variadas, usando protocolos específicos entre uma comunicação fim a fim. Muitos desses fluxos possuem um comportamento previsível, por exemplo, a transmissão de um arquivo via FTP (*File Transfer Protocol*), ou uma mensagem de e-mail. Porém, um *streaming* de vídeo tem características que o difere dos demais fluxos: a ocupação de grande quantidade de banda, e a possível variação de cenas com consequente aumento (ou diminuição) na quantidade dos *bytes* transmitidos.

As aplicações para transmissão de *streaming* de vídeo (YouTube, Vimeo, NetFlix, Telecine Play, e etc.) tem despertado o interesse da comunidade científica quanto ao comportamento das redes. Salgueiro (2004) alertava sobre um conjunto de diferentes requisitos (disponibilidade do canal de comunicação, atraso na propagação, erros dos quadros, entre outros) não necessariamente convergentes, durante a transmissão de dados, que precisa ser considerado quando se trata de uma mesma estrutura em rede de computadores. Esses requisitos tem motivado pesquisadores quanto a qualidade dos serviços prestados por ISP (*Internet Service Provider*): a experiência do usuário em serviços de dados, voz e vídeo.

Há trabalhos sobre QoE (*Quality of Experience*) bem específicos, tais como: (Youtube (FELD et al., 2011), *buffers* (HOHLFELD et al., 2014), Internet das Coisas (MEHMOOD et al., 2012)); porém, há quem busque compreender a relação entre QoS (*Quality of Service* – Qualidade de Serviço) e QoE - *Survey* em máquinas de aprendizagem (AROUSSI; MELLOUK, 2014), estimativa do QoE baseado em parâmetros de QoS (RIVERA et al., 2013), estudo de caso da correlação QoS-QoE (OROSZ et al., 2014).

Na expectativa de encontrar uma metodologia que representasse a experiência do usuário tal qual como o vivencia durante uma transmissão de um fluxo de informações qualquer (por exemplo, vídeo), Alreshoodi e Woods (2013) analisaram oito métodos com os quais procuraram demonstrar as influências de alguns parâmetros de QoS, desde a camada física até a camada de aplicação, e verificaram o quão difícil é relacioná-las entre si. Ou seja, problemas específicos requerem soluções específicas, por não serem capazes de prever toda a subjetividade inerente ao ser humano quanto às suas experiências. Concluíram, ainda, que há a necessidade da compreensão dos aspectos dessa correlação, bem como as suas influências (ALRESHOODI; WOODS, 2013).

A experiência do usuário apresenta dados relevantes que demandam atenção. De acordo com a pesquisa feita pelo Instituto Brasileiro de Pesquisas (IBOPE, 2013), “o Brasil ocupa a terceira posição em quantidade de usuários ativos na Internet (52,5 milhões), enquanto que no primeiro e segundo lugares estão Estados Unidos (198 milhões) e Japão (60 milhões), respectivamente”. Todavia, em outro levantamento feito pelo Centro Regional para o Desenvolvimento da Sociedade da Informação, foi identificada a seguinte situação: a média de assinantes que utilizam até 256Kbps gira em torno de 22%; e até 1Mbps - 14,2% (em relação a taxa de *download*), ou seja, quase 40% desses mais de 50 milhões de usuários possuem banda larga com até 1Mbps entre as cinco regiões do Brasil (CETIC.BR, 2014).

Com base nessa média de largura de banda contratada, Tanenbaum(2011) considera que, a essa mesma taxa, é possível assistir um vídeo pela Internet, utilizando compressão de dados, com qualidade razoável (TANENBAUM; WETHERALL, 2011). Se o usuário desejar assistir filmes em alta definição (HD - *High Definition*) nessa mesma taxa, provavelmente não terá uma experiência agradável.

Neste sentido, as seguintes perguntas norteiam a presente pesquisa: como correlacionar a experiência do usuário com base na arbitragem de parâmetros de QoS em ambiente controlado durante a transmissão de um *streaming* de vídeo, e como essa arbitragem, de fato, manterá essa experiência com qualidade? Como as características de equipamento desse usuário impactará a sua experiência, mesmo que as condições da rede sejam favoráveis? Não se deve desconsiderar a influência da aplicação na escolha do protocolo que irá transportar esses fluxos.

1.1 Problemática e Hipótese

Durante a fase de investigação dos trabalhos relacionados, buscou-se encontrar pesquisas que utilizassem técnicas de classificação de pacotes através do método de Serviços Diferenciados em ambiente real. Porém, daqueles em que a busca encontrou ocorrência, o uso de modelos matemáticos e softwares simuladores foram os escolhidos. A seguir é feito um breve descritivo dos trabalhos encontrados.

O modelo proposto por Hsu et al. (2010) simula uma rede com roteadores de borda – cujos pacotes são classificados e condicionados para o tráfego – e roteadores de núcleo da rede, que tem a tarefa de, apenas, encaminhar os pacotes classificados baseados em seus comportamentos por salto, associados a um DSCP (*DiffServ Code Point*) (HSU et al., 2010, p. 183). O ambiente de testes usado foi um computador com sistema operacional Windows XP e linguagem de programação C++ (HSU et al., 2010, p. 189).

Varela e Laulajainen (2011) tratam das questões de VoIP (*Voice over Internet Protocol*), cuja temática principal é demonstrar a aplicação de estimativas de QoE para melhorar soluções de gerenciamento de mobilidade a nível de rede existentes. Foi utilizado o *software* comercial Birdstep¹ (não foi referenciado no trabalho do autor).

E, ao abordar a correlação entre as métricas de QoS e QoE sobre IPTV (*Internet Protocol TeleVision*) usando redes com serviços diferenciados, Botia (2012) comparou, a partir do *software* NS-2 (*Network Simulator*), métricas de qualidade objetiva e subjetiva com base em um conjunto de redes configuradas com parâmetros de QoS em serviços diferenciados e melhor esforço, e concluiu que deve se dar atenção especial aos parâmetros de codificação de vídeo, dando destaque às empresas de telecomunicações quanto à oferecer alta qualidade em serviços multimídia.

Há outros trabalhos que envolvem QoS e QoE, mas o objetivo para essa problematização é trazer a tona questões sobre os caminhos que alguns autores seguiram para comprovarem as suas hipóteses. Este trabalho, em particular, busca um olhar ainda não pensado, ou mesmo publicado, a partir de outra perspectiva da mesma problemática.

Testes iniciais, em rede local, demonstraram haver problemas na reprodução de filme em *FULL HD*, tanto em conexões cabeadas, quanto sem fio. Ao ajustar o QoS da rede,

¹<http://birdstep.com/>

houve melhora significativa. Ajustando os parâmetros no *player*, adicionou-se melhoria na experiência do usuário. Esses testes ocorreram a partir de um servidor de *streaming* e um dispositivo final.

O intento da pesquisa foi correlacionar a experiência do usuário à parâmetros de qualidade nos serviços de rede e a influência dos dispositivos finais, criando um ambiente controlado que permitiu uma avaliação mais próxima das condições de um ambiente real, no caso, a Internet, a partir de dispositivos finais dos próprios usuários.

1.2 Objetivos da Dissertação

O objetivo principal desta dissertação é correlacionar os parâmetros de QoS da rede aos recursos de *hardware* nos dispositivos dos usuários que afetam a sua experiência durante uma transmissão de fluxos multimídia.

1.2.1 Objetivos Específicos

Os seguintes objetivos específicos foram traçados:

- Identificar qual(ais) parâmetro(s) de QoS representa(m) a melhor experiência para o usuário baseado na tecnologia de Serviços Diferenciados (em inglês, *Differentiated Services*);
- Avaliar as condições perceptuais do usuário quanto a sua experiência durante uma transmissão de fluxos multimídia;
- Integrar o *hardware* NetFPGA (*Network Field Programmable Gate Array*) com os roteadores Cisco para simular uma nuvem Internet sem marcação de pacotes.

1.3 Material e Métodos

Para compor um processo de pesquisa sobre a Qualidade de Experiência de um usuário baseado nos parâmetros de QoS, faz-se necessário recorrer a uma revisão sistemática dos autores que tratam, validam ou questionam sua problemática.

Com base nos pressupostos teóricos, foi criado um ambiente de testes que permita avaliar diversos cenários, e a sua correlação com a rede de computadores. Utilizar-se-á a plataforma NetFPGA e roteadores Cisco como recursos necessários para a adaptação de um dos modelos propostos por Alreshoodi e Woods (2013).

Espera-se, com esse processo de pesquisa, estudar, analisar, discutir e identificar a melhor forma de avaliação da experiência do usuário, conforme a sua percepção, durante uma transmissão de fluxos multimídia.

Recursos

Para a realização dos experimentos desta dissertação, foram necessários os seguintes recursos:

- **Placa NetFPGA:** *hardware* para pesquisas em redes de computadores.
- **Sistema Operacional Linux versão CentOS 7 64 bits:** baseado em código aberto.
- Software para gerar e analisar tráfego na rede;
 1. **iPerf:** gerador de tráfego capaz TCP (*Transport Control Protocol*) e UDP (*User Datagram Protocol*).
 2. **Cacti:** Ferramenta para análise de desempenho da rede.
- Software para simular *streaming* de vídeo.
 1. **VLC:** reprodutor multimídia livre, de código aberto, multi-plataforma, e um arcabouço que reproduz a maioria dos arquivos de mídia, bem como DVD, CD de áudio, VCD e vários protocolos de fluxo de rede (OUESTAD, 2015).

A placa NetFPGA foi adquirida via PROJETO DO EDITAL UNIVERSAL do CNPq, processo **474313-2012-3** (ORDONEZ, 2014). Os *softwares*, são *OpenSource*, e o PC necessário para trabalhar com o NetFPGA foi adquirido com recursos próprios.

O ambiente de testes foi realizado no Instituto Federal de Sergipe (IFS) e na Universidade Federal de Sergipe (UFS), ambos campus São Cristóvão.

1.4 Contribuições da Dissertação

Nesta dissertação, obteve-se as seguintes contribuições:

- Demonstrar o impacto do fluxo multimídia em roteadores que não possuam QoS aplicado;
- Demonstrar o benefício da aplicação de QoS numa transmissão de fluxo multimídia através da técnica de serviços diferenciados em ambiente concorrente com fluxos diversos;
- Apresentar as funcionalidades do *hardware* NetFPGA, projetos desenvolvidos pela comunidade, e as limitações entre as versões disponíveis quanto aos projetos já consolidados;
- Avaliar a experiência do usuário, correlacionado-a às características dos parâmetros de qualidade de serviços de rede e dos dispositivos dos usuários.

1.5 Organização da Dissertação

O texto desta dissertação foi organizado em 6 capítulos, que forneceu toda a base conceitual e empírica para o entendimento completo sobre como a experiência do usuário é afetada ao ajustar alguns parâmetros da camada de rede - atraso, variação e perda de pacotes. Os tópicos a seguir descrevem os conteúdos dos capítulos.

- O Capítulo 2 fundamenta conceitos relacionados às redes de computadores, e apresentou os principais trabalhos sobre QoS e QoE;
- As características sobre a placa NetFPGA são tratadas no Capítulo 3;
- O Capítulo 4 demonstra a integração entre a placa NetFPGA e roteadores Cisco;
- Após toda a fundamentação teórica, experimentos e análises dos resultados são tratados no Capítulo 5;
- E, finalmente, o Capítulo 6, as principais contribuições e sugestões para possíveis trabalhos futuros.

Capítulo 2

Identificação de serviços da camada de rede que afetam a experiência do usuário

Uma prestação de serviços, quando baseada em acordo pré estabelecido, exige que a contratada cumpra com os termos acordados. A contratante, por sua vez, utiliza tais serviços, avaliando-os conforme as cláusulas contratuais. Se alguma dessas condições não for cumprida, os termos do acordo podem ser revistos para ajustar à nova realidade, ou simplesmente aplicado alguma penalização pelo descumprimento do acordo.

Em redes de computadores, esses termos de acordo para a execução de serviços de infraestrutura são regulados a partir de especificações padronizadas, cujo objetivo é garantir que um pacote encaminhado de uma origem qualquer chegue ao seu destino com as mesmas características que o originou, em prazos aceitáveis, aceitando ou não, variações de tempo na entrega dos pacotes, a depender do tipo de acordo estabelecido e, podendo ainda, descartá-los, no caso em que não se espere fluxos de dados fora dos termos acordados.

Uma vez estabelecido esse acordo, o usuário final é afetado positiva, ou negativamente, em sua experiência, e essas condições tem características bastante subjetivas – humor, equipamento, infraestrutura, e etc. – que afetam diretamente na avaliação do resultado de um serviço prestado.

As seções a seguir apresentam as características que afetam os serviços de rede e a forma como o usuário percebe o resultado desses termos, a partir de documentos submetidos a comitês que avaliam as especificações que podem se tornar um padrão para acesso à Internet, com qualidade e de maneira previsível.

2.1 Qualidade de Serviços – QoS

A RFC (*Request for Comment*) 791, que especifica o Protocolo de Internet (*Internet Protocol*), no detalhamento das operações do protocolo, destaca os 04 mecanismos chaves para o fornecimento dos serviços da camada de rede: Tipo de Serviço (ToS - *Type of Service*), Tempo de Vida (TTL - *Time to Live*), Opções (*Options*) e *checksum* do cabeçalho (*Header Checksum*). O campo ToS é o campo usado para indicar o QoS desejado. Tipo de Serviço é uma abstração de um conjunto de parâmetros genéricos que caracterizam a escolha de serviços fornecidos em redes que fazem a Internet (AGENCY, 1981). Esse campo do cabeçalho IP será objeto de ajustes dos parâmetros de QoS.

A camada de rede permite a transferência de pacotes entre origem e destino, podendo passar por diversos roteadores (*hops* – saltos) que possam existir no caminho a ser percorrido. Em termos de serviços dessa camada, o modelo de referência TCP/IP procura fazer com que esses mesmos pacotes sejam entregues com o “menor esforço possível” (*Best Effort*). Existem duas formas de implementar os serviços dessa camada: orientado e não-orientado a conexões. Apesar da Internet adotar serviços não-orientados a conexões, Tanenbaum e Wetherall (2011) fazem uma ressalva quanto a esse serviço. Eles informam que a Internet está evoluindo, através da Comutação de Rótulos Multiprotocolo (MPLS - *MultiProtocol Label Switching*) e redes locais virtuais (VLAN - *Virtual Local Area Network*), para adoção de serviços orientados a conexões.

Há quatro aspectos que devem ser resolvidos quanto à Qualidade de Serviço, conforme (TANENBAUM; WETHERALL, 2011), sendo eles:

1. Que aplicações de rede são necessárias;
2. Como regular o tráfego que entra na rede;
3. Como reservar recursos nos roteadores para garantir o desempenho, e;
4. Se a rede pode aceitar mais tráfego com segurança.

Nenhuma técnica isolada lida, de modo eficaz, com todos esses aspectos.

Dentre os principais parâmetros de QoS – latência (*delay*), variação (*jitter*), largura de banda e perda de pacotes, ou de sequência – latência e *jitter* são os que mais influenciam na

transmissão de fluxos multimídia. Aplicações multimídia podem requerer sincronismo nas transmissões de vídeo e áudio, caso seja um serviço de vídeo sob demanda. Enlaces com alta densidade de fluxos de dados geralmente provocam congestionamentos, com consequentes perdas de pacotes nos roteadores. Em conexões UDP, pelo fato de não ser um protocolo orientado a conexão, não há confirmações do que foi entregue ao destinatário. Essa característica prejudicará serviços multimídia que utilizem *codecs* (*Encoder/Decoder*) que necessitem a chegada de pacotes na ordem em que foram gerados; do contrário, a apresentação do conteúdo não refletirá o que foi originalmente criado.

O mecanismo usado para controle da qualidade de serviço em redes de computadores é a técnica de enfileiramento dos pacotes. A Figura 2.1 apresenta a atuação desta técnica desde a chegada do pacote até o seu encaminhamento para o destino. Após a chegada do pacote, há uma classificação que poderá ser feita por meio de uma Lista de Controle de Acesso (ACL - *Access Control List*), ou configurando algum mecanismo de fila, no exemplo da figura, foi usado o Controle de QoS Modular (CISCO, 2011). Se o pacote corresponder ao critério estabelecido na regra, será colocado na fila com sua respectiva prioridade. Caso contrário, o pacote segue seu caminho sem alterações, respeitando a ordem de chegada. No caso de dois pacotes com prioridades diferentes serem encaminhados para a mesma interface de saída, sairá primeiro o de maior prioridade.

A subseção a seguir detalhará as técnicas de enfileiramento usados nos equipamentos Cisco, mas não se trata de tecnologia proprietária, segue a recomendação do IETF (*Internet Engineering Taks Force* – Força Tarefa de Engenharia para Internet).

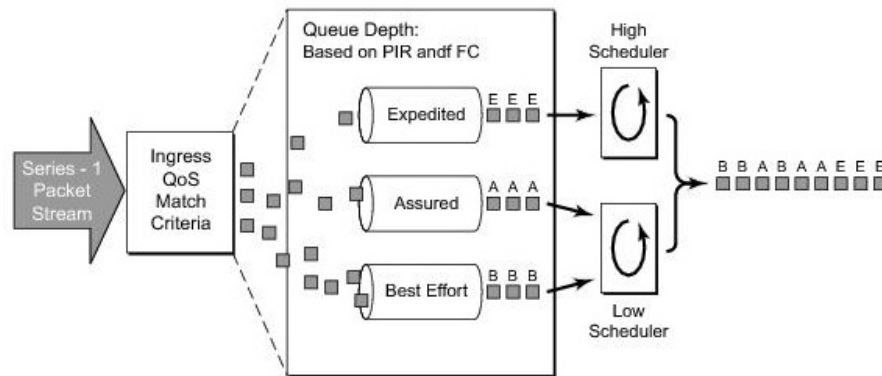
2.2 Técnicas de Enfileiramento

Para (HERSENT; GURLE; PETIT, 2002, p. 290), enfileiramento é a “arte de se administrar filas congestionadas, de modo a garantir determinada largura de banda, latência e taxa de perda de pacotes aos diversos fluxos através dos nós”.

Há duas categorias que podemos agrupar essas técnicas, são elas:

- Ordenação dos pacotes nas filas de saída
 - **FIFO** – *First In, First Out*, apenas encaminha na ordem em que os pacotes

Figura 2.1: Modelo de enfileiramento de tráfego para 3 filas e 3 classes



Fonte: (ALCATEL-LUCENT, 2012)

chegam.

- **Baseada em classe**, atribuem um índice de prioridade a cada pacote recebido, colocando-os em uma fila lógica.
- **Algoritmos de enfileiramento justo**, utiliza algoritmos de enfileiramento justo
 - * (FQ - *Fair Queueing*) ou balanceado (WFQ - *Weighted Fair Queueing*);
 - * *CBWFQ*, Enfileiramento justo ponderado com base em classe.
- **Gerenciamento das perdas de pacotes**, permite combinação com as técnicas de enfileiramento.
 - Sobrecarga simples;
 - Detecção antecipada aleatória e balanceada.

Cada técnica tem seu algoritmo de controle das filas, sendo que os serviços da camada de rede mais usados são: Integrados e Diferenciados. Nosso objeto de estudos se restringirá a explicar à arquitetura deste último serviço, no qual será abordado na próxima subseção.

As palavras-chave “DEVE”, “NÃO DEVE”, “REQUERIDO”, “DEVERÁ”, “NÃO DEVERÁ”, “DEVERIA”, “NÃO DEVERIA”, “RECOMENDADO”, “PODE”, e “OPCIONAL”, nos documentos das RFC (*Request For Comment*), devem ser interpretados como descrito em (BRADNER; UNIVERSITY, 1997).

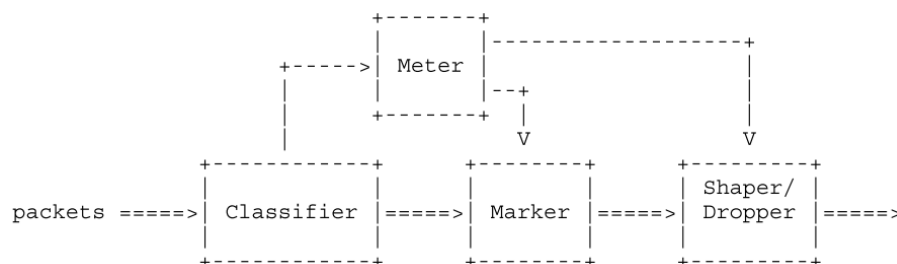
Serviços Diferenciados – DiffServ

A RFC 2475 a definiu como “uma arquitetura a ser implementada de modo escalável na Internet” (BLAKE et al., 1998, p. 1). Sua escalabilidade é obtida através da agregação de tráfego, a partir da classificação da marcação do pacote no campo do cabeçalho IPv4 ou IPv6, denominado DSFIELD (*DiffServ Field*). Esta implementação se dá em nós da rede, incluindo funções para:

- encaminhamento dos pacotes por salto;
- classificação de pacotes, e;
- condições do tráfego, incluindo métricas, marcação, modelagem e políticas.

Todas essas funções precisam condicionar o tipo de tráfego da rede em si quanto à qualidade do serviço oferecido. A arquitetura prevê essas funcionalidades, na qual ficou definido como “Condicionadores de Tráfego”. Este poderá conter os seguintes elementos: um “medidor” (*meters*), um “marcador” (*markers*), um “calibrador” (*shapers*) e um “descartador” (*droppers*). A Figura 2.2 mostra o diagrama de funcionamento do condicionador e classificador de tráfego. Vale ressaltar que nem sempre o pacote passará por todas as etapas. Para que isso ocorra, basta que o pacote não possua marcação explícita, percorrendo, apenas, as etapas de classificador e marcador (implicitamente, todo pacote não classificado receberá uma marcação e classificação com valores *default*).

Figura 2.2: Visão lógica de um classificador de pacotes e Condicionador de Tráfego



Fonte: (BLAKE et al., 1998)

A seguir, o detalhamento da função de cada um desses classificadores e condicionadores de tráfego (BLAKE et al., 1998, p. 16).

Meters, medem propriedades temporais do fluxo de pacotes selecionados por um classificador, conforme um perfil de tráfego especificado.

Markers, configura o campo DS do cabeçalho IP de um pacote para um *codepoint* em particular, adicionado o pacote marcado a um DS agregado.

Shapers, atrasa um, ou todos, os pacotes de um fluxo de tráfego, de modo a fazer com que o fluxo esteja em conformidade com o perfil do tráfego.

Droppers, descarta um, ou todos, os pacotes de um fluxo de tráfego, de modo a fazer com que o fluxo esteja em conformidade com o perfil do tráfego.

A arquitetura trata de muitas outras questões de rede, tais como comportamento por salto (PHB - *Per-Hop Behaviors*) ((BLAKE et al., 1998), p. 19), *Network Resource Allocation* (1998, p. 20), *Interoperability with Non-Differentiated Services-Compliant Nodes* (1998, p. 25), *Multicast Considerations* (1998, p. 26) e *Security and Tunneling Considerations* (1998, p. 27).

A seguir, abordaremos as classes de serviços que compõem o *DiffServ*.

RFC 3246 – Encaminhamento Expedido (EF)

Comportamento por salto (PHB) é uma técnica utilizada para controlar atraso, descartes de pacotes e *jitter*, e a RFC 3246 (2002) trata a forma como esses pacotes são encaminhados após a sua entrada, via interface de rede, em dispositivos que implementem a camada 3 do modelo de referência para Interconexão de Sistemas Abertos (OSI - *Open Systems Interconnect*). Seu objetivo é fornecer uma base para serviços que tratem esses parâmetros, porém sem detalhar as implementações, uma vez que essa etapa é responsabilidade dos fabricantes de dispositivos.

Em *links* de redes de área ampla (WAN - *Wide Area Network*), o atraso de propagação e as filas em *switches* e roteadores são as causas dos atrasos em redes de pacotes. Uma vez que atraso de propagação é uma propriedade fixa da topologia, atraso e variação são minimizados quando atrasos na fila são minimizados. Diante do exposto, *jitter* pode ser definido como a variação entre o atraso máximo e mínimo (DAVIE et al., 2002, pag. 2).

Definição Formal

Um nó que suporta EF (*Expedited Forwarding*) em uma interface I configurada a uma taxa R DEVE satisfazer as seguintes Equações (DAVIE et al., 2002, p. 5-6):

$$d_j \leq f_j + E_a \quad \text{para todo } j > 0 \quad (2.1)$$

onde f_j é iterativamente definida por: (considerando $f_0 = 0$, $d_0 = 0$)

$$f_j = \max(a_j, \min(d_{j-1}, f_{j-1})) + l_j/R, \quad \text{para todo } j > 0 \quad (2.2)$$

Para essa definição, temos:

- d_j , é o tempo em que o último *bit* do j -ésimo pacote EF para partir, realmente, deixa o nó da interface I ;
- f_j , é o momento da partida que o j -ésimo pacote EF sai de I , tempo “ideal” de saída, ou antes que o último *bit* do pacote deixe o nó;
- a_j , é o tempo que o último *bit* do j -ésimo pacote EF destinado à saída I atual chega ao nó;
- l_j , é o tamanho (em *bits*) do j -ésimo pacote a partir de I ;
- R , é a taxa EF configurada na saída I (em *bits/segundo*);
- E_a , é a relação de erro para tratamento de pacote individual EF;
- d_0 e f_0 , não se referem a partida real do pacote, mas são usados especificamente para propósitos de recursão. O tempo inicial deverá ser escolhido tal que nenhum pacote esteja no sistema no tempo ZERO.
- Para as definições de a_j e d_j , o “último *bit*” do pacote inclui o *trailer* da camada 2, se estiver presente, porque um pacote, geralmente, não pode ser considerado disponível para a transmissão de dados até que um *trailer* seja recebido.

Além disso, um nó que suporte EF numa interface I , a alguma taxa configurada R , DEVE satisfazer as seguintes Equações (DAVIE et al., 2002, p. 6-7):

$$D_j \leq F_j + E_p \text{ para todo } j > 0 \quad (2.3)$$

onde F_j é iterativamente definida por: (considerando $F_0 = 0$, $D_0 = 0$)

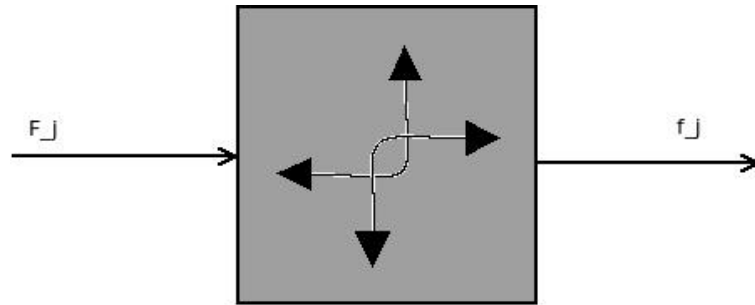
$$F_j = \max(A_j, \min(D_{j-1}, F_{j-1})) + L_j/R, \text{ para todo } j > 0 \quad (2.4)$$

- D_j , é o tempo atual de partida do pacote individual EF que chegou ao nó destinado à interface I no tempo A_j ;
- F_j , é o momento de partida do pacote individual EF que chegou ao nó destinado à interface I no tempo A_j ;
- A_j , é o tempo que o último *bit* do j -ésimo pacote EF destinado a chegar à saída I chegue ao nó;
- L_j , é o tamanho (em *bits*) do j -ésimo pacote a chegar ao nó que está destinado à saída I ;
- R , é a taxa EF configurada na saída I (em *bits/segundo*);
- E_p , é a relação de erro para tratamento do EF agregado;
- D_0 e F_0 , não se referem a partida real do pacote, mas são usados especificamente para propósitos de recursão. O tempo inicial deverá ser escolhido tal que nenhum pacote esteja no sistema no tempo ZERO.
- Para as definições de A_j e D_j , o “último *bit*” do pacote inclui o *trailer* da camada 2, se estiver presente, porque um pacote, geralmente, não pode ser considerado disponível para a transmissão de dados até que um *trailer* seja recebido.

É o fato de D_j e F_j se referirem aos tempos de partida para o j -ésimo pacote a chegar que faz as equações 2.3 e 2.4 cientes da identidade de pacotes. Esta é a diferença fundamental entre as duas últimas equações das duas primeiras.

Na Figura 2.3, o instante que um pacote chega no roteador (F_j , Equação 2.2), e o pacote deixando o roteador (f_j , Equação 2.4).

Figura 2.3: Representação do Encaminhamento Expedido



Fonte: Elaborado pelo autor.

Atraso (*delay*) e Variação (*jitter*)

Dado um valor conhecido de E_p e um conhecimento dos limites no tráfego EF oferecido a uma dada interface de saída, somada ao longo de todas as interfaces de entrada, é possível limitar o atraso e instabilidade que serão experimentados pelo tráfego EF deixados do nó através dessa interface. O atraso limite é definido como (DAVIE et al., 2002, p. 8-9):

$$D = B / R + E_p \quad (2.5)$$

onde:

- R é a taxa de serviço EF configurada na interface de saída;
- a carga total oferecida pelo tráfego EF destinada a interface de saída, somadas ao longo das interfaces de entrada, é limitada pelo “*token bucket*” de taxa $r \leq R$ e profundidade B .

Uma vez que o atraso mínimo através do dispositivo é de, pelo menos, zero, D também fornece um limite de *jitter*. Para proporcionar um limite no *jitter*, o valor de E_p para um dispositivo PODE ser especificado como dois componentes separados, de tal modo que

$$E_p = E_{\text{fixo}} + E_{\text{variavel}} \quad (2.6)$$

onde E_{fixo} representa o atraso mínimo que pode ser experimentado por um pacote EF através do nó.

Perda (Loss)

Se considerarmos uma carga do tráfego EF suficientemente alta (por exemplo: grandes rajadas inesperadas de qualquer interface de entrada de uma única vez), um dispositivo com *buffer* finito poderá perder pacotes. Assim, deve ser possível determinar se um dispositivo está em conformidade com a definição de EF, mesmo quando alguns pacotes são perdidos. Isto é feito através da realização de um teste “*off-line*” de conformidade com as Equações 2.1 a 2.4. É assumido que, pacotes que não deixam o nó, são considerados perdidos e retirados do fluxo de entrada. Os pacotes restantes passam a constituir o fluxo de chegada (variável a_j), e os pacotes que deixaram o nó constituem o fluxo de partida (variável d_j). A conformidade com as equações pode, portanto, ser verificada, apenas, considerando os pacotes que passaram, com sucesso, através do nó.

A explanação sobre o mecanismo de Encaminhamento Expedido se faz necessário porque o experimento se baseou nos parâmetros de QoS que tratam, especificamente, de fluxos de dados. A seguir será apresentado o mecanismos baseado em Encaminhamento Assegurado.

RFC 2597 – Encaminhamento Garantido (AF)

Um serviço que assegure o encaminhamento dos pacotes é definido na RFC 2597 (HEINANNEN et al., 1999, p. 2) como sendo “um meio para um provedor de domínio DS (*Differentiated Service*) oferecer diferentes níveis de garantia de encaminhamento dos pacotes recebidos de um domínio DS cliente”. Foram definidas quatro classes AF (*Assured Forwarding*), na qual cada uma está presente em nó DS, sendo reservado uma certa quantidade de recursos para o encaminhamento (*buffer* e largura de banda).

Cada pacote IP é marcado, de acordo com o cliente ou provedor de domínio DS, com um dos três possíveis valores de precedência para descarte. Em caso de congestionamento, o descarte de um pacote será determinado por sua precedência dentro da classe na qual foi atribuída. Essa política será muito útil para classificar pacotes que não sejam *streaming* de vídeo.

Portanto, em um nó DS, o nível de garantia de encaminhamento de um pacote IP, depende de (i) o quanto o encaminhamento de recursos foi alocado para a classe AF na qual o pacote pertence, (ii) qual é a carga atual da classe AF, e, em caso de congestionamento dentro da

classe, (iii) qual é a precedência de descarte do pacote.

Filas e Comportamento de Descarte

A RFC 2597 (HEINANEN et al., 1999, p. 4) recomenda sete especificações para o tratamento de pacotes marcados com classe AF. Apresentaremos as que se aproximam das características de tráfego concorrente com fluxos multimídia para melhor encaminhamento desse tipo de pacote.

- A especificação prevê que uma implementação AF DEVE tentar minimizar congestionamentos de longo prazo dentro de cada classe, ao passo que permita congestionamento de curto prazo resultante de rajadas. É necessário que se implemente um algoritmo de gerenciamento de fila ativo, sendo o descarte antecipado aleatório (RED - (*Random Early Drop*)) um exemplo;
- Permitir que o AF PHB seja utilizado em diferentes ambientes operacionais, os parâmetros de controle do algoritmo de descarte DEVEM ser configuráveis independentemente, para cada pacote com precedência de descarte, e para cada classe AF.

Os valores recomendados dos *codepoints* AF são:

Class 1	Class 2	Class 3	Class 4
AF11 = 001010,	AF21 = 010010,	AF31 = 011010,	AF41 = 100010,
AF12 = 001100,	AF22 = 010100,	AF32 = 011100,	AF42 = 100100,
AF13 = 001110,	AF23 = 010110,	AF33 = 011110,	AF43 = 100110.

Dessa maneira, uma implementação, cuja classe seja configurada com alta precedência de descarte em pacotes que não sejam um *streaming* de vídeo, garantirá um encaminhamento com maior prioridade durante um transmissão desse tipo de fluxo de dados.

Na seção sobre os roteadores Cisco, serão tratadas as formas de como os dispositivos se comportarão desde a chegada de um pacote numa interface I , até a sua saída em uma das interfaces disponíveis.

2.3 Qualidade de Experiência – QoE

A Qualidade de Experiência pode ser medida em todas as camadas do modelo OSI, como fora abordado na Introdução. De acordo com a definição do Setor de Padronização das Telecomunicações (ITU - *International Telecommunication Union*), QoE é “a aceitabilidade geral de uma aplicação, ou serviço, tal como percebida subjetivamente pelo usuário final” (DVORAK, 2007). E faz duas observações:

- A Qualidade de Experiência inclui o efeito completo dos sistemas fim-a-fim (clientes, terminais, redes, serviços de infraestrutura, e etc);
- A aceitabilidade geral pode ser influenciada pela expectativa e contexto do usuário.

O mais comum é controlar os parâmetros da camada de rede (largura de banda, atraso, perda e variação) para evitar que o usuário receba um fluxo qualquer com baixa qualidade durante a sua reprodução. Como será apresentado na seção seguinte, a subjetividade na percepção do usuário é um fator determinante em sua experiência, a exemplo de como avaliar a sua satisfação no envolvimento com determinado sitio, ou sua diversão num jogo em tempo real (BROOKS; HESTNES, 2010).

Há duas abordagens metodológicas para avaliar essa correlação: avaliação Objetiva e Subjetiva (ALRESHOODI; WOODS, 2013). Essas técnicas, separadamente, não avaliam a experiência do usuário. As tabelas 2.1 e 2.2 demonstram o que é avaliado na forma objetiva e subjetiva, respectivamente.

2.4 Trabalhos Correlatos

Os modelos avaliados por Alreshoodi e Woods (2013) foram:

Tabela 2.1: Medidas Objetivas
Exemplo de medidas objetivas de desempenho do usuário

Medidas	Interação Pessoa-Conteúdo	Interação Pessoa-Pessoa
Precisão do usuário na conclusão da tarefa	Reconhecimento correto da informação visual	A medida em que um problema compartilhado é corretamente resolvido
Número de entradas do usuário	Clicks do mouse	Palavras faladas
Número de interrupções	Interrupções de conversação com um avatar	Interrupções de conversação com outra pessoa

Dados parcialmente extraídos de (BROOKS; HESTNES, 2010).

Tabela 2.2: Medidas Subjetivas
Exemplo de medidas subjetivas de desempenho do usuário

Medidas	Interação Pessoa-Conteúdo	Interação Pessoa-Pessoa
Efetividade	Avaliação de informações na precisão visual recebida	Avaliação da extensão em que foi alcançado o objectivo da comunicação
Satisfação	Avaliação de satisfação global com o conteúdo	Avaliação de satisfação global com a comunicação
Presença social	Avaliação se um avatar parece real	Avaliação da impressão de contato pessoal com outras pessoas

Dados parcialmente extraídos de (BROOKS; HESTNES, 2010).

- *IQX hypothesis (exponential interdependency of quality)*: baseado numa fórmula genérica, na qual os parâmetros de QoS e QoE estão conectados através de um relacionamento exponencial (FIEDLER; HOSSFELD; TRAN-GIA, 2010). O resultado obtido com o experimento não considerou a variação da natureza do IP em relação aos impactos da qualidade como é percebida pelo usuário, e o *software* de emulação de rede (NIST Net), para efeito de caracterização da perda de pacotes, não consegue gerenciar essas perdas, simulando-as de maneira aleatória, não refletindo o “*modus operandi*” em ambiente real.
- *VQM (Video Quality Metric)-based Mapping Model*: função n -dimensional do QoS (onde “ n ” é o número de diferentes parâmetros de QoS) (PINSON; WOLF, 2004). O foco desse estudo foi somente em parâmetros de QoS na camada de rede. De acordo com Alreshoodi e Woods (2013, p. 58), será necessário um melhor ajuste na curva da função logarítmica para melhorar a relação entre QoS/QoE.
- *QoE Model using Statistical Analysis method*: técnica de modelagem estatística empregada, na qual correlaciona-se parâmetros de QoS com estimativas de percepções de QoE, e identifica o grau de influência de cada um dos parâmetros de QoS sobre a percepção do usuário (AROUSSI; MELLOUK, 2014). Eles destacam que, ao envolver outros fatores relacionados ao conteúdo (parâmetros de codificação de vídeo), podem maximizar a qualidade percebida pelo usuário, e alcançar uma utilização mais eficiente da rede.
- *QoE Models based on Machine Learning methods*: propõe uma nova abordagem precisa e adaptativa para a construção de modelos de previsão de QoE, usando algoritmos de classificação em máquina de aprendizagem, com dados treinados para testes subjetivos (KATSARAKIS; THEODOSIADIS; PAPADOPOULI, 2014).

- *QoE model using Crowdsourcing for subjective tests*: baseado na plataforma *Microworkers*, permite a condução de *surveys* de testes *online*, tal com o YouTube (FELD et al., 2011). Eles descobriram que a idade do expectador e o conteúdo do vídeo não impactam significativamente a experiência do usuário, mas se a reprodução do vídeo congelar, ou sofrer alterações por mais que três segundos em dois momentos aleatórios, esses tenderão a ficar bastante insatisfeitos com o resultado.
- *QoE model using a Resource Arbitration System*: fundamentado na integração entre as camadas de Rede e Aplicação (NQoS e AQoS) (SILLER; WOODS, 2003), a variação dos parâmetros das camadas de rede e aplicação são ajustados manualmente para alcançar a QoE. O foco da pesquisa foi em ajustes dos parâmetros de QoS da camada de rede.
- *QoE model considering equipment and environment factors*: a partir do sinal de origem, interferências no ambiente *wireless* (taxa de error de quadro, atraso e variação) podem ocorrer. Diferentes parâmetros de QoS podem ser aplicados ao equipamento do usuário. Em seguida, ruídos, tremores e luz ambiente formam a interferência ambiental, na qual diferentes parâmetros de QoE são usados (HAN et al., 2012). Entendem que há muito o que melhorar quanto ao algoritmo de escalonamento QBS (*QoE Based Scheduling*), necessitando de testes mais elaborados, em diversos cenários, ajustando outros parâmetros de QoS;
- *QoE model based on Quantitative and Qualitative Assessment*: uma combinação de abordagens qualitativas e quantitativas. Foi utilizada a técnica RST (*Rough Set Theory*) para a avaliação quantitativa, enquanto que o *framework* CCA (*Catalog, Categorize, and Analyze*) para avaliação qualitativa (ALRESHOODI; WOODS, 2013). Quanto à avaliação quantitativa, eles aprenderam que os diferentes tipos de conteúdo de vídeo exigem diferentes níveis de suporte QoS. No entanto, a influência dos fatores ambientais tendem a influenciar no mundo real, a compreensão da interdependência entre si torna-se muito complexa.

É mister que a metodologia para avaliar a experiência do usuário trate dos requisitos de QoS. Dos trabalhos pesquisados sobre a relação entre QoS e QoE que influenciam

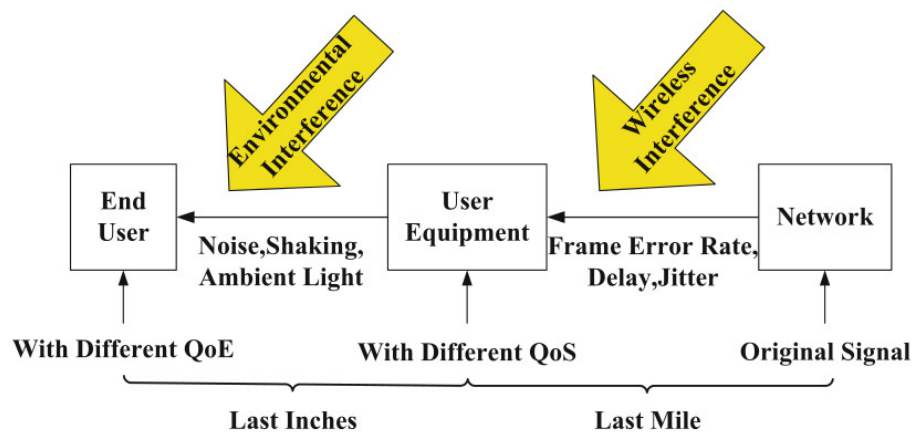
nessa experiência, todos trataram essas questões, na maior parte, entre a camada de rede e a de aplicação. Questões da camada física também influenciam diretamente no resultado após uma transmissão, a exemplo de erros de CRC *Cyclic Redundancy Check* durante uma comunicação de voz sobre IP (VoIP).

As próximas duas subseções a seguir abordaram dois modelos que avaliaram a qualidade de experiência com base nas camadas de rede e aplicação, bem como os dispositivos finais e características do ambiente na qual a comunicação aconteceu, uma vez que a avaliação dessa experiência ocorreu em infraestrutura para ambiente sem fio.

Modelo considerando equipamentos e fatores ambientais

No modelo proposto por Han et al. (2012), a base da pesquisa é demonstrar que não adianta baixar a qualidade da origem de um fluxo (áudio e/ou vídeo) para resultar numa melhor experiência para o usuário final. A ideia é testar os parâmetros de *hardware* e ambiente relacionado ao equipamento do usuário para, em seguida, ajustar a demanda da qualidade do sinal da rede, de modo a agrupar as necessidades dos usuários em seus diferentes níveis. O modelo usado é apresentado na Figura 2.3.

Figura 2.4: Modelo QoE considerando fatores do ambiente e equipamento



Fonte: (HAN et al., 2012)

A partir desse modelo, eles criaram uma função matemática para expressar essa experiência com base nos parâmetros de QoS, equipamento do usuário (UE - *User Equipment*) e fatores do ambiente. A equação é apresentada a seguir:

$$L_{QoE}^{i,u} = f_{QoE}^u(L_{QoS}^{i,u}, L_{EI}^{i,u}, L_{EQ}^{i,u}) \quad (2.7)$$

donde $L_{QoE}^{i,u}$ representa o usuário final “i” com nível de serviço de QoE “u”. $L_{QoS}^{i,u}$ representa os parâmetros quantitativos da rede, $L_{EI}^{i,u}$ reflete a interferência do ambiente do usuário, $L_{EQ}^{i,u}$ o parâmetro quantitativo referente ao equipamento do usuário. f_{QoE}^u é uma função que mapeia o nível de QoE a partir do QoS, equipamentos e fatores de ambiente.

Essa função será importante para delimitar quais parâmetros serão utilizados para proporcionar a melhor experiência do usuário na transmissão de um *streaming* de vídeo no experimento. Por exemplo, se, após os testes, os parâmetros ajustados que resultaram a melhor experiência em determinado fluxo de dados foram atraso e perda de pacotes, pode-se montar a função QoS como sendo:

$$L_{QoS}^{i,u} = f_{QoS}^u(P_{QoS}^{i,d}, P_{QoS}^{i,l}) \quad (2.8)$$

onde $P_{QoS}^{i,d}$ representará o atraso (*delay*), e $P_{QoS}^{i,l}$, a perda de pacotes (*loss*). Para representar as funções em relação aos equipamentos e fatores de ambiente, basta substituir os índices e seus respectivos parâmetros.

O trabalho se propôs a apresentar um novo escalonador baseado em QoE, chamado de QBS (*QoE Based Scheduling*), comparando o desempenho em relação ao algoritmo PFS (*Proportional Fair Scheduling*) para melhorar o uso dos recursos da rede (HAN et al., 2012, p. 605). Os testes de *downlink* foram feitos sobre o protocolo CDMA2000 EV-DO Rev.B 3-Carrier system, com 19 células e 57 setores incluídos, cujas estações base estavam distante entre si por 2.500 metros, com os usuários uniformemente distribuídos. Não ficou claro se os testes foram em ambiente real ou simulado, apesar da seção ter como título “*Simulation Results*”.

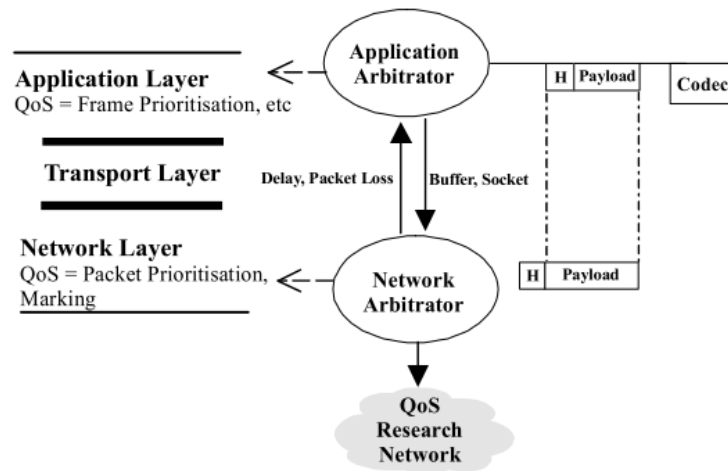
Modelo baseado na Arbitragem de Recursos

O desafio enfrentado por Siller e Woods (2003) foi tentar comprovar que, ao controlar os parâmetros que influenciam o QoS, tais como atraso, perda e *jitter*, é possível alcançar uma melhor qualidade de experiência do usuário. Para tal, eles propuseram um modelo experimental inserido em um *framework* QoE, na qual a interação do QoS com a rede e a camada

de aplicação são considerados como um todo, não como entidades individuais. A relação da QoE com a camada é definida como alguma função de fatores da rede e aplicação, definido com NQoS (*Network QoS*) e AQoS (*Application QoS*) (SILLER; WOODS, 2003, p. 1). Para completar o ambiente de testes, foi adicionado um mecanismo de arbitragem para gerenciar os mecanismos de QoS, sendo estes direcionados através de uma tabela de referência para QoS, com pesos para refletir o atraso, a variação e a perda de pacotes.

A Figura 2.4 representa como ocorre essa interação entre as camadas. O QoS é gerenciado e controlado em cada camada por uma entidade denominada como “camada de arbitragem” (SILLER; WOODS, 2003, p. 2).

Figura 2.5: Modelo QoE baseado na arbitragem de recursos



Fonte: (SILLER; WOODS, 2003)

A métrica geral de QoS da rede (NQoS) definida no âmbito QoE é considerada como a soma dos fatores ponderados, dadas por três métricas de rede: atraso, *jitter* e perda de pacotes. Para esse propósito, o árbitro de rede toma conhecimento do tipo de mídia a ser transmitido. O árbitro gerencia os mecanismos de QoS, as políticas definidas e fatores de ponderação iniciais utilizando as tabelas de QoS RT¹ (*Reference Table*). Fatores de ponderação são entidades dinâmicas, influenciadas por flutuações de QoS da rede e requisitos de QoS de aplicação. Para efeito dos seus estudos, o QoE será considerado normalizado para o coeficiente com valor máximo igual a 1 (SILLER; WOODS, 2003, p. 4).

¹Os autores não apresentaram as tabelas no artigo publicado.

O passo-a-passo para a arbitragem de QoS na rede acontece da seguinte maneira: as métricas de QoS são medidas e inseridas no arbitrador da rede; em seguida, este aplica os fatores de ponderação iniciais conforme tabelas de referências de QoS, sendo a métrica de QoS calculada. O fator de ponderação com o maior valor ditará o mecanismo de QoS a ser implementado (na primeira consulta à tabela), a exemplo do *DiffServ* ou *IntServ*. Então, o QoE é medido conforme definido na tabela de referência de QoS. Se nenhuma métrica for calculada, adota-se o valor 0,5 para refletir a influência máxima sobre o QoE. As variações de medidas de QoS (ΔQ) são inseridas no arbitrador, os valores ponderados são ajustados com os novos valores calculados e, finalmente, repita o processo desde o início para melhorar o QoE.

Dentre as equações propostas por Siller e Woods (2003, p. 4), o coeficiente (β_i) na equação (2.9)

$$\beta_i = \Delta Q_i - \Delta Q_{i-1} \quad (2.9)$$

melhorará a qualidade de experiencia do usuário quando for positivo, e piorará se tiver seu valor negativo.

2.5 Considerações finais do Capítulo

Esse capítulo apresentou uma fundamentação teórica sobre os mecanismos de funcionamento para garantir a Qualidade de Serviços da camada de rede do modelo de referência OSI. As técnicas de priorização de pacotes abordadas foram restringidas aos serviços diferenciados, cujos modos de encaminhamento dos pacotes adotam as técnicas de Encaminhamento Expedido e Garantido. Essa teoria ajuda na compreensão em como proporcionar a melhor experiência do usuário com base em ajustes dos parâmetros de QoS, sendo que este pode ser aplicado em todas as camadas do modelo OSI. Cabe ressaltar que cada camada da arquitetura OSI terá protocolos correspondentes ao nível da camada a ser avaliada/mensurada, que, como já foi afirmado na Introdução deste trabalho, a subjetividade inerente ao ser humano torna qualquer avaliação complexa, na medida em que se queira avaliar a relação entre as camadas da pilha do modelo de referência OSI para garantir a melhor experiência de um usuário qualquer.

Esses aspectos contextualizam e direcionam a dissertação, ampliando a percepção sobre a importância no uso de *hardware* específico para redes de computadores, que serão tratados no capítulo 3 deste trabalho.

Capítulo 3

Integração entre Placa NetFPGA e Roteadores Cisco

Este capítulo descreve as características de um dispositivo desenvolvido para servir de plataforma em pesquisas de redes de computadores, capaz de proporcionar aos pesquisadores, flexibilidade aliada a máximo desempenho, e a sua integração com dispositivos de fabricantes consolidados no mercado mundial.

3.1 O modelo NetFPGA-1G-CML

A NetFPGA é um *hardware* de arquitetura aberta, flexível e com desempenho na velocidade nominal das interfaces de rede. Existem mais de 150 instituições, num total acima de 40 países, que utilizam-na em pesquisas para Redes de Computadores (NETFPGA.ORG, 2014a).

A NetFPGA vem evoluindo desde a sua concepção no ano de 2001, sendo disponibilizado 3 versões de placa: NetFPGA 1G-CML, NetFPGA 10G e NetFPGA SUME. A versão NetFPGA 1G foi descontinuada (NETFPGA.ORG, 2014a). A seguir será apresentado o modelo NetFPGA -1G-CML¹

Desenvolvida para operar sobre uma interface PCIe (*Peripheral Component Interconnect express*) 4x segunda geração, possui 4 portas *Gigabit Ethernet*, incorporando o FPGA

¹As informações sobre o NetFPGA estão disponíveis em <https://github.com/NetFPGA/NetFPGA-public/wiki/NetFPGA-1G-CML-Reference-Manual>. Acesso em 18/02/2015.

(*Field Programmable Gate Array*) Kintex-7 325T, da Xilinx. Essa plataforma foi projetada para dar suporte a arquitetura NetFPGA de Stanford, com projetos de referência disponíveis através do (*NetFPGA GitHub Organization*). É totalmente compatível com o software Xilinx Vivado e a suíte ISE Design², bem como projetos de softwares embarcados Xilinx SDK (*Software Development Kit*). A placa tem as seguintes características, conforme a Figura 3.1 (NETFPGA.ORG, 2014b):

Figura 3.1: Placa NetFPGA-1G-CML



Fonte: (NETFPGA.ORG, 2014b)

- Lógica *Field Programmable Gate Array* (FPGA);
- Quatro portas de rede Gigabit Ethernet;
- *Quad Data Rate Static Random Access Memory* (QDRII + SRAM) - 36 MBit (4,5 MBytes);
- *Double-Data Rate Random Access Memory* (DDR3 DRAM), com capacidade de 512 MBytes;
- PCI Express Gen. 2;
- Armazenamento em cartão SD e memória *flash* BPI (*Byte Peripheral Interface*)

²Nesse caso, as instruções de configuração da placa sugere a instalação da suíte ISE *Suite Design* versão 14.6 (já existe versão mais recente - 14.7), da Xilinx. Há forte recomendação para uso do VIVADO, não há mais suporte ao ISE.

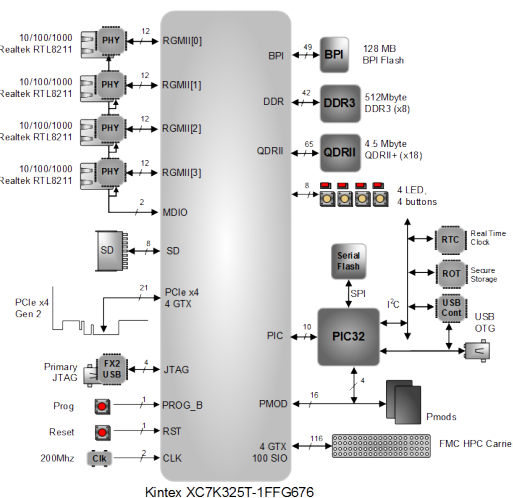
- Interface de expansão (conectores FMC e PMOD);
- Características adicionais, tais como microcontrolador PIC e USB, RTC, chip com autenticação Crypto;
- Padrão PCIe *Form Factor*;
- Flexível, código aberto.

Configuração do FPGA

Toda a lógica de programação do sistema é armazenada em memória SRAM. Por ser um tipo de memória volátil, o dispositivo é configurado toda vez que é energizado. A configuração dos dados é conhecida como *bitstream*, cujos formatos usados são “bit” ou “mcs”. Pode ser configurado via *flash BPI*, *flash drive USB off-board* ou via PC.

A Figura 3.2 apresenta a arquitetura do FPGA Kintex-7.

Figura 3.2: Arquitetura Kintex XC7K325T-1FFG676



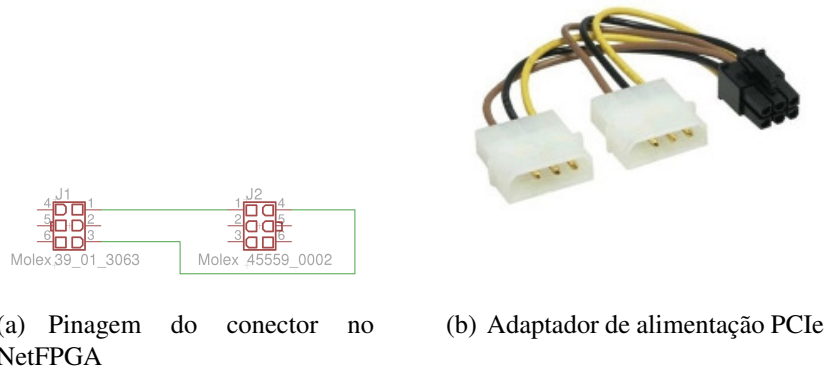
Fonte: (NETFPGA.ORG, 2014b)

Energização da Placa

A placa requer as tensões de 5V e 12V. Sua alimentação é via conector Molex J17 da placa, permitindo usá-la sem a necessidade de estar conectada a uma placa-mãe via barramento

PCIe. Esse conector não acompanha o kit da placa NetFPGA, mas pode ser encontrado no mercado. A Figura 3.3 mostra a pinagem e o adaptador, respectivamente ³.

Figura 3.3: Características de energização da placa



Fonte: (NETFPGA.ORG, 2014b)

Osciladores e Relógios

Um oscilador a 125 MHz com baixo *jitter* é fornecido para uso das interfaces *Ethernet*, e um *drive* oscilador de 50 MHz para a configuração do *clock* principal do FPGA, além de um oscilador diferencial de 200 MHz para fornecimento de um *jitter* ultra baixo.

Memórias

A seguir daremos uma breve descrição das principais funcionalidades de cada uma das memórias presentes na placa NetFPGA.

FPGA

Inclui blocos de 4096 *bytes* com dois *bit* de correção de erro, totalizando 1,78 MB de correção de erros *on-chip* para permitir uma grande variedade de propósitos para armazenamento de programas em aplicações para sistemas embarcados do tipo “*bare metal*”.

³Disponível em: <http://lista.mercadolivre.com.br/cabo-adaptador-pcie#D5BA:cabo-adaptador-pcie,B:55D>. Acesso em 22/02/2015.

DDR3

Inclui um *chip* Micron MT41K512M8 512 MB SDRAM DDR3 que emprega um barramento de dados a 800 MHz capazes de operar a uma taxa de 1600 MT/s (*MegaTransfer*) de dados transferidos tanto na borda de subida, quanto na de descida. Possui um Gerador de Interface de Memória (MIG - *Management Interface Generator*), cuja função é dar assistência na geração de interface para selecionar os tipos de peças e configurar recursos de E/S (Entrada/Saída) FPGA para a interface de memória de *hardware*.

QDRII+

Fornecida para aplicações que exigem alta velocidade e memória de baixa latência. As aplicações comuns incluem *buffers* FIFO e pesquisas de tabela. A noção de taxa de dados “*Quad*” vem da capacidade de ler simultaneamente a partir de uma porta de leitura unidirecional, e gravar em uma porta de escrita unidirecional em ambas as bordas do relógio. É capaz de operar a até 450 MHz para produzir taxas de transferência de dados da porta até 900 MT/s por 2 *bytes*. Isso produz uma largura de banda máxima de até 3.6 GB/s. O núcleo gerador Xilinx é capaz de gerar e configurar uma interface de camada física para a QDRII+ via assistente Gerador de Interface de Memória.

Flash BPI

Memória *flash* em uma configuração x16 128 MB, é fornecida para suportar a configuração FPGA de alta velocidade após um *reset*, tem capacidade suficiente para armazenar várias configurações de dispositivo.

Cartão SD

Suporta um segundo recurso de armazenamento não-volátil, que também é removível. Este conector suporta um cartão de memória SD (*Secure Digital*) de tamanho padrão, e cumpre todas as exigências da camada física de ambos os protocolos de barramento SPI (*Serial Peripheral Interface*) e SD.

Interfaces de comunicação

Interface PCIe

Quatro, dos oito transceptores GTX de alta velocidade, são implementações dedicadas em até quatro canais da segunda geração PCIe (5 GB/s) na comunicação com o processamento do sistema em um *host*.

Interface PHY (Física)

Quatro transceptores Realtek RTL8211 Ethernet (Phys) são fornecidos como interface para conexões de rede via conectores RJ-45 *on-board*. O comportamento padrão de auto-negociação permite que cada PHY se ajuste independentemente a sua taxa de dados, a 10/100 Mbps ou 1 Gbps, conforme necessário. Os dados são transferidos de e para os PHYs através de uma (RGMII - *Reduced Gigabit Media Independent Interface*).

Subsistema PIC

Microcontrolador para o gerenciamento de OTG USB (*Universal Serial Bus*), relógio em tempo real, e interconexão de armazenamento seguro. O PIC é pré-programado com código de teste de produção e uma capacidade para carregar *bitstreams* FPGA a partir de um cartão de memória USB. É possível reprogramar o PIC para suportar aplicações de usuários finais que fazem uso de vários outros recursos do subsistema de PIC.

E/S On-board

Built-in on-board I/O, inclui quatro LED e seis botões. Quatro desses botões são de uso geral, os outros dois são reservados para funções especiais. Os botões vermelhos são para funções especiais reservadas para o uso como um *reset on-chip* (BTN4 - RESET) para redefinir a lógica do projeto, e uma redefinição de configuração (BTN5 - PROG), que inicia uma nova sequência de configuração do FPGA como o que ocorre no *power-on*.

Conectores de Expansão

Conector PMOD

Acessórios que suportam uma grande variedade de interfaces externas que aumentam a flexibilidade do sistema. Os sinais lógicos não são pares combinados; eles são encaminhados sem controle de impedância ou correspondência de atraso.

Conector FMC

Inclui um conector de transporte VITA-57 compatível com FMC (FPGA *Mezzanine Card*). Um conector Pin (HPC) de contagem alta é utilizado para fornecer o máximo de compatibilidade com uma variedade de placas de *mezanino* disponíveis comercialmente.

Para que essa plataforma opere conforme especificações do fabricante, é necessário a execução de uma série de requerimentos disponíveis em uma comunidade virtual para esse fim. As recomendações de *hardware* e *software* estão disponíveis na comunidade GitHub sobre NetFPGA.

Nesse sitio, há outras informações relevantes, tais como FAQ (*Frequently Asked Questions*), organização e uso da base do código em NetFPGA, bibliotecas IP, dentre outros tópicos.

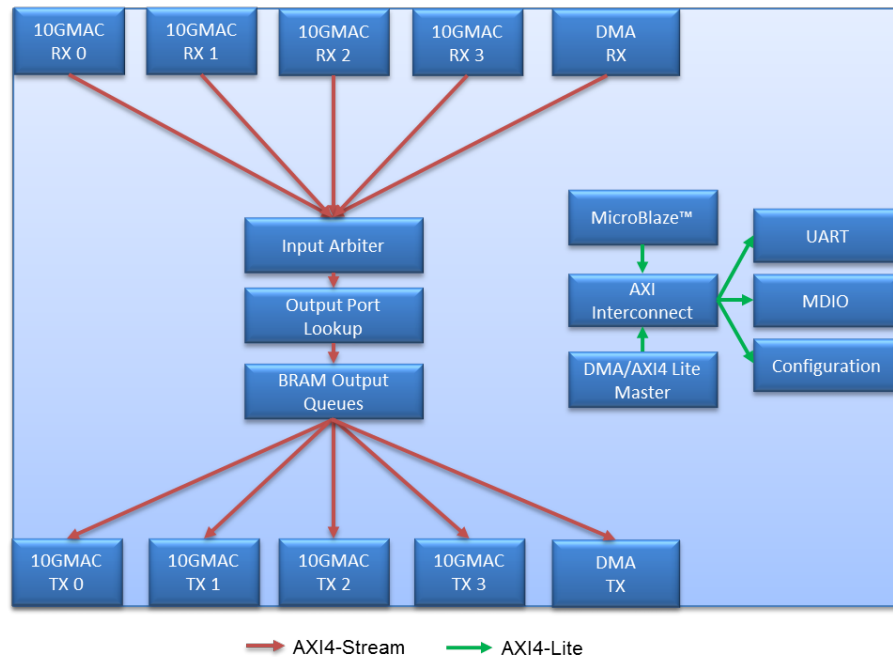
3.2 Arquitetura da placa

O diagrama da Figura 3.4 representa o funcionamento, em modo *pipeline*, da placa como uma interface de rede (NIC - *Network Interface Card*).

Os pacotes chegam através do módulo da interface de rede, sendo transformados para sinais XGMII (o 'X' dependerá da versão da placa), em seguida são lidos pelo Xilinx 10G MAC (*Media Access Control*) (esse diagrama foi usado para a versão 10G, mas é compatível com a 1G-CML) para, finalmente, serem transformados em fluxo AXI4. O lado **TX** faz o mesmo, mas em sentido oposto.

O módulo de fila da interface RX se conecta ao módulo de entrada do arbitrador (*input arbiter*), este contém cinco interfaces de entrada: quatro módulos de interface 1G e um módulo de DMA (*Direct Memory Access*). Cada módulo de entrada do arbitrador se conecta a fila de entrada, que implementa escalonador FIFO. Um esquema de pesquisa nas entradas

Figura 3.4: Diagrama de funcionamento do projeto NIC



Fonte: (NETFPGA.ORG, 2014a)

das filas caracteriza um gerenciamento tipo *Round Robin*, selecionando a cada momento uma fila não-vazia, escrevendo um pacote completo a partir dele para o próximo estágio no caminho de dados, ou seja, o módulo *output port lookup*.

O módulo *output port lookup* é o responsável por decidir qual porta um pacote deverá sair. Tomada a decisão, o pacote é manipulado para o módulo da fila de saída. Um esquema básico é implementado para envio e recebimento dos pacotes de/para a CPU (*Central Processor Unit*).

Uma vez que o pacote chega no módulo da fila de saída, já terá um destino marcado. De acordo com o destino, entrará numa fila de saída dedicada. Essas filas são organizadas intercaladamente, seguindo uma ordem: uma porta *Ethernet* física, uma porta DMA, e assim sucessivamente. Filas pares são atribuídas às portas físicas *Ethernet*, enquanto que as ímpares, atribuídas às portas DMA virtuais.

Os projetos que permitem a placa ter outras funções são inseridos entre o módulo *input arbiter* e *output port lookup*, sendo necessário muita atenção nas ligações lógicas do novo módulo, de modo a não alterar o fluxo descrito anteriormente.

Projetos NetFPGA

Durante a fase de instalação do *software* da placa, no sitio da comunidade de desenvolvedores de projetos para uso da placa, a disparidade no quantitativo de projetos desenvolvidos na primeira versão da placa em relação às outras é enorme, sendo esta já depreciada, tendo um total de 53 projetos para a versão 1G, contra apenas 6 projetos para a versão NetFPGA-CML usada no ambiente de testes dessa dissertação. Já a versão 10G possui 16 projetos, e a última versão – NetFPGA-SUME – ainda não constam projetos compartilhados.

Devido a flexibilidade de uso da placa, esta foi configurada no modo *Reference Router*.

Um ponto a ser destacado é o fato da incompatibilidade de versões do módulo FPGA entre si. A placa NetFPGA-1G é totalmente incompatível com todas as demais, obrigando ao desenvolvedor reescrever o código para a FPGA a ser utilizada. O sitio⁴, guia *Documentation*, contém todas as FPGA disponíveis, e a série 7 só é compatível com outros de mesma ordem (XILINX, 2014, p. 27), apesar de o sitio⁵ afirmar que são compatíveis, conforme lista de FPGA a seguir.

1. NetFPGA-1G FPGA Logic

- Xilinx Virtex-II Pro 50

2. NetFPGA-1G-CML

- Xilinx Kintex-7 325T

3. NetFPGA-10G

- Xilinx Virtex-5 TX240T

4. NetFPGA-SUME

- Xilinx Virtex-7 690T

Outra questão é quanto a linguagem de programação. Por se tratar de programação em *hardware*, há a necessidade de bons conhecimentos em arquitetura de computadores em baixo nível, o que torna a ferramenta mais difícil em seu uso pela comunidade acadêmica que não tenha formação em Ciência/Engenharia da Computação.

⁴<http://www.xilinx.com/support.html>

⁵https://github.com/NetFPGA/NetFPGA-public/wiki/Home_NetFPGA-1G-CML

3.3 Roteadores Cisco modelo 1841

A Cisco Systems, fabricante de dispositivos para Redes de Computadores, fundada em 1984, é uma das principais líderes mundiais nesse segmento da Informática. A sua grande contribuição para o desenvolvimento de padrões para a Internet é comprovada através dos muitos documentos submetidos ao IETF, tais como os padrões **EF** 2002 e **AF** 1999, objetos de estudos desta pesquisa. Uma lista de compatibilidade das suas soluções com os padrões OSI (*Open System Interconnection*) podem ser consultadas em Cisco (2011).

Esse modelo foi escolhido por ser um roteador modular para serviços integrados (ISR - *Integrated Services Router*) com interfaces LAN (*Local Area Network*) e WAN (*Wide Area Network*) configuráveis. Possui duas portas FE (*Fast Ethernet*), porta console e auxiliar de até 115,2 kbps (*kilo bits per second*). Sua memória SDRAM (*Synchronous Dynamic Random Access Memory*) tem a função de armazenar as configurações durante a operação do equipamento, bem como as tabelas de roteamento, além de servir de *buffer* dos pacotes que chegam através das interfaces de rede.

Na próxima subseção serão apresentadas as configurações utilizadas para a aplicação do QoS nas interfaces dos roteadores.

Configuração do roteador – para *streaming* e fluxos do *iPerf*

Os parâmetros de configuração usados para permitir a aplicação do QoS nas interfaces dos roteadores encontram-se no apêndice.

As classes foram criadas para classificar os pacotes que chegam através das interfaces, por meio de uma lista de controle de acesso que dá permissão aos pacotes através de protocolos. No caso do experimento, o IP foi utilizado para classificar os endereços de *host* com destino ao servidor de *streaming*. Uma vez identificado o tipo de fluxo, chega o momento de marcar o pacote através de uma política, também aplica às interfaces do roteador. A diferença entre classificar e marcar um pacote é que, na classificação, a política é aplicada no ingresso do pacote; conseqüentemente, a marcação acontecerá na saída da interface. A título de esclarecimento: especificamente em roteadores Cisco modelo 1841, a memória é compartilhada em ambos os sentidos, mas há modelos em que o processamento acontece por porta, aumentando a capacidade de encaminhamento dos pacotes.

As configurações variam de acordo com o que se deseja alcançar em termos de resposta. Se a intenção for priorizar o encaminhamento dos pacotes, a política a ser aplicada fará com que a interface aloque o máximo da largura de banda que destinado àquela política; caso seja uma garantia de largura de banda mínima, há um outro conjunto de instruções para que o roteador defina um mínimo de banda para àquele fluxo em específico.

Assim, a flexibilidade de opções para tratamento dos fluxos dá ao usuário condições de fazer ajustes específicos a cada classe criada, em cada salto por onde o pacote passar, mas o DSCP é um padrão que não exige configuração em todos os saltos tal como o RSVP (*Resource Reservation Protocol*), que exige um circuito virtual que garanta uma largura mínima da banda.

3.4 Integração entre NetFPGA e Cisco

A Placa NetFPGA, em modo *router*, tem duas formas de configuração: *router kit* – a partir de uma interface gráfica que permite adicionar parâmetros de rede, tais como endereço MAC e IP – e SCONE (*Software Component Of NEtFPGA*). O primeiro, apesar da interface gráfica desenvolvida em linguagem de programação JAVA, não executou o projeto conforme esperado. Neste, é necessário criar um arquivo com as informações das interfaces de rede a serem utilizadas, com os respectivos endereços MAC e IP em modo CLI (*Command Line Interface*) *shell*⁶ para se criar o processo *daemon* sendo executado em *background*. Este, ao ser iniciado, cria a tabela de roteamento atribuídas as interfaces de rede pré-configuradas, disparando pacotes com tamanhos aleatórios entre si.

Um ponto importante desse processo: não há necessidade de manter o processo executando indefinidamente, até porque, depois de determinado tempo, ele “trava” ao encher o *buffer*, não mais permitindo o encaminhamento de novos pacotes. Mesmo configurando as interfaces nas configurações de rede do Linux, e inserindo manualmente os endereços MAC, a placa só permitiu encaminhar os quadros quando o *software* SCONE foi executado. O motivo é que, até o momento antes dessa execução, a tabela MAC não está devidamente criada na memória da placa NetFPGA. A consequência é que, mesmo a interface respondendo ao parâmetro ICMP (*Internet Control Message Protocol*) através do comando *ping*, só foi

⁶Passo a passo disponível em <https://github.com/NetFPGA/NetFPGA-public/wiki/NetFPGA-1G-CML-Reference-Router>

possível entre o hospedeiro da placa NetFPGA e o *host* de destino. Se uma estação de outra rede tentasse enviar um ICMP passando pelas interfaces da NetFPGA em modo roteador, a ausência dessa tabela impedia que a origem alcançasse o seu destino.

Por fim, mas não necessariamente o fechamento sobre as limitações de compatibilidade da placa, o protocolo de roteamento OSPFv2 (*Open Shortest Path First*) – parte do projeto de referência *Reference Router* – não foi compatível com o padrão da RFC 2328. Ao configurar o roteador para trabalhar com esse protocolo, apesar de haver o reconhecimento do protocolo, a mensagem reproduzida no programa de captura de pacotes *Wireshark* (a captura estava sendo executada na interface da placa NetFPGA) apresentava a mensagem “*BAD LENTGTH*”, inviabilizando seu uso. A ideia era simular a queda/congestionamento de um *link* para observar se o caminho alternativo seria usado com base nos critérios do algoritmo de roteamento dinâmico.

Em testes simulados por outros pesquisadores (com vídeos apresentados em canais do YouTube), esse algoritmo só funcionou entre placa NetFPGA do mesmo modelo.

3.5 Considerações finais do Capítulo

Nesse capítulo, a necessidade de se descrever os *hardwares* utilizados no experimento se fez para que os resultados realizados se aproximassem ao máximo de um cenário real. O uso da placa NetFPGA tem a função principal de representar os “n” saltos que possam existir entre um provedor de serviços de Internet até uma residência de um usuário final. Como em boa parte dos serviços de acesso à Internet não contemplam quaisquer mecanismos de priorização de pacotes, a placa NetFPGA entra com a finalidade de permitir tráfego diverso, bem como de não fazer qualquer marcação de pacotes, ao passo que os roteadores Cisco devem ser configurados para demonstrar os benefícios na oferta de serviços diferenciados. No capítulo a seguir, será detalhado o ambiente de testes e como foi criada a linha de base do experimento.

Capítulo 4

Ambiente de testes

O processo de preparação do ambiente levou em consideração a escolha dos equipamentos (dispositivos intermediários e finais), e o Sistema Operacional (SO). Quanto aos dispositivos intermediários, os testes foram iniciados com o software de emulação de redes GNS3, mas substituído posteriormente por dois roteadores Cisco modelo 1841 para representar um cenário mais próximo de um ambiente real em serviços de Internet. O SO padrão foi Linux, com variações nas distribuições: CentOS 7, Ubuntu 16.04, Fedora 22 e Linux Mint 17.3. As ferramentas utilizadas para a análise do tráfego de rede foram o Cacti versão 0.8.8b, *software freeware* amplamente utilizado em pesquisas acadêmicas e empresas corporativas e *Wireshark*, cuja característica é permitir uma análise mais específica sobre as características individuais das camadas do modelo de referência TCP/IP. A escolha do Cacti foi por sua flexibilidade em obter “traps” SNMP (*Simple Network Management Protocol*), e por possuir uma série de MIB (*Management Information Base* – Base de Informações para Gerenciamento) para os mais variados dispositivos de rede.

Quanto aos usuários finais, não houve essa preocupação, uma vez que a ideia é mensurar a experiência com o mínimo de interferência em relação ao dispositivo usado. Porém, foi necessário coletar as especificações técnicas dos dispositivos finais para compreender as causas determinantes das condições perceptuais dos usuários.

Todo o cenário de testes foi realizado no Instituto Federal de Sergipe, no Campus São Cristóvão.

Com relação ao *link*, como o canal de comunicação é de 100 Mbps, não houve necessidade de reajuste da velocidade nas interfaces dos roteadores para simular um canal WAN,

porque os vídeos exigiram largura de banda suficiente para suportar as rajadas que determinadas cenas provocavam, chegando a picos de 27 Mbps por dispositivo. Como foi previsto um cenário com cinco dispositivos finais, a taxa de transmissão chegou a aproximadamente 75 Mbps (valor arredondado) na saída do roteador de borda, lado cliente, provocando o congelamento da imagem em alguns dispositivos, de modo aleatório.

Inicialmente, o indicador de consumo da banda foi de, aproximadamente, 2,5 Mbps durante a transmissão, em alta definição, do filme *Insurgente*, com resolução de 1.920 x 800 ppi (*pixels per inch*) a taxa de 23,976 quadros por segundo.

A seguir será apresentado o processo de preparação do ambiente de testes.

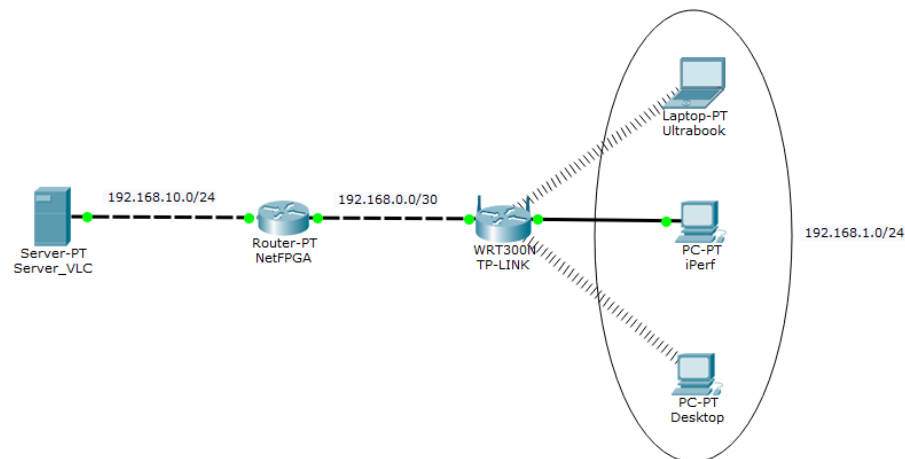
4.1 Ambiente para uso da placa NetFPGA

A NetFPGA é uma placa com plataforma aberta. Por ser um dispositivo altamente flexível, exige do pesquisador conhecimentos especializados em programação na linguagem padrão HDL (*Hardware Description Language*), limitando, assim, o seu uso para àqueles que não a dominam. Primeiro: a comunidade científica desenvolveu poucas soluções para simulação de testes em redes de computadores, em particular para a versão 1G-CML (modelo adquirido pelo programa de mestrado da instituição); segundo: as funções implementadas se restringem a usá-la como interface de rede, comutador ou roteador, sendo esse último com o algoritmo de roteamento OSPF implementado em nível de S.O. (Obs: nos testes com o uso do OSPF em conjunto com os roteadores Cisco, há um erro no tamanho do pacote que impede a comunicação entre a placa NetFPGA e os roteadores Cisco via esse protocolo, mas, ainda assim, se fez necessário executá-lo para que a tabela ARP (*Address Resolution Protocol*)) registrasse os endereços MAC dos dispositivos que encaminhavam os quadros através da placa.

Porém, uma nova topologia foi criada para que o experimento utilizasse a placa NetFPGA, conforme a Figura 4.1. Todas as interfaces de rede dessa placa operam, exclusivamente, a *Gigabit Ethernet*. Para integrar a placa ao cenário proposto, foi adquirido um aparelho de multipropósito da fabricante TP-LINK, modelo Archer C2, que possui todas as interfaces *Gigabit Ethernet*; sua conexão *wireless* é padrão 802.11ac, podendo alcançar um *throughput* de até 750 Mbps. Dessa forma, foi possível realizar os testes com a placa Net-

FPGA, mas de maneira independente do que fora previsto inicialmente no projeto, sem, no entanto, invalidar o objetivo dessa pesquisa.

Figura 4.1: Topologia usando a placa NetFPGA e gerador de tráfego iPerf



Fonte: Elaborado pelo autor.

4.1.1 A escolha do Hardware

A única observação necessária quanto a escolha da placa-mãe é que seja adquirida de fabricantes já consolidados no mercado internacional (ASUS, MSI, GIGABYTE) porque, ao adquirirmos um computador de mesa (fabricação nacional) para uso nos testes, mesmo não havendo problemas na instalação do Sistema Operacional e realizado os procedimentos de sintetização do projeto da placa NetFPGA, o barramento PCIx da placa não foi reconhecido pelo SO Linux CentOS 7, não carregando o *driver* das interfaces de rede da placa. Porém, recomenda-se a aquisição de um processador com bom poder de processamento, porque, a depender do tamanho do projeto, o tempo de sintetização poderá chegar a horas para conclusão. Numa rápida comparação, os tempos de sintetização obtidos nas configurações de placa-mãe/CPU (a quantidade de memória RAM mínima recomendada é de 4 GB) foram os seguintes:

- Placa-mãe ASUS modelo A55BM-E/BR e Processador AMD A4 4000 Dual-Core 3.0 GHz - aproximadamente 1h30min;

- Placa-mãe Login modelo H81H3M4 e processador Intel Core i3 - aproximadamente 45 minutos.

OBS: o disco rígido (fabricante Seagate, 500 GB e 7200 RPM) foi o mesmo para ambas as placas, com nova instalação do S.O. em ambos os casos.

4.1.2 A escolha do Sistema Operacional

Os procedimentos para instalação e configuração da placa encontram-se no site¹ Sistema Operacional Fedora 20 kernel 3.14.17. Para acesso ao software da placa, se faz necessário registrar-se na comunidade do *gitHUB*. Apesar do passo-a-passo estar bem explicado, ocorreram vários erros durante a execução dos procedimentos recomendados, principalmente no momento de *synetizar* o projeto (procedimento que configura o módulo FPGA da placa em relação a blocos lógicos configuráveis, de entrada e saída, e conexões (também conhecido como roteamento). Após várias tentativas frustradas para operacionalizar o ambiente, ficou decidido testar outras versões de distribuições Linux, e a mais estável recomendada em fóruns é a distribuição CentOS, que atualmente está instalada a versão 7 (1511), kernel 3.10.0-327.18.el7.x86_64. A recomendação é instalar a versão na língua inglesa, porque a quantidade de pacotes que se fez necessário adicionar ao escolher a opção em português demonstrou que há incompatibilidade entre os idiomas.

4.2 Configuração das classes de serviços nos roteadores Cisco

A qualidade de serviço do tráfego de rede para pacotes que passam pelos roteadores Cisco foram ajustados com base na prioridade de encaminhamento dos pacotes. A outra forma de controle dos fluxos é com base na largura de banda. A razão da escolha por prioridades se dá por permitir gerenciar atrasos e *jitter*, e melhoria no tempo de resposta da aplicação. O gerenciamento por controle de banda é mais indicado para *links* WAN com baixa taxa de transmissão.

¹<https://github.com/NetFPGA/NetFPGA-public/wiki/NetFPGA-1G-CML-Getting-Started-Guide>

Porém, não quer dizer que aplicações, com necessidade de prioridade em seus fluxos, não se beneficiem de um QoS que reserve um mínimo de largura de banda. Ao contrário, implementações que usam essa técnica garantem máxima largura de banda. O mecanismo para tal garantia possui, internamente em filas específicas para pacotes com prioridade, um *token bucket* que mede a carga oferecida, garantindo que o fluxo esteja em conformidade com a taxa configurada do *link*. Ou seja, apenas tráfegos que estejam em conformidade com o *token bucket* tem garantia de baixa latência. Em outras palavras, qualquer tráfego excedente somente será enviado se o *link* estiver com baixa ocupação; do contrário, os pacotes serão descartados, afim de evitar congestionamento do canal (Cisco Systems, 2008).

Todo roteador Cisco mantém dois conjuntos de filas, conforme Quadro 4.2:

Figura 4.2: Quadro representando os conjuntos de filas nos roteadores Cisco

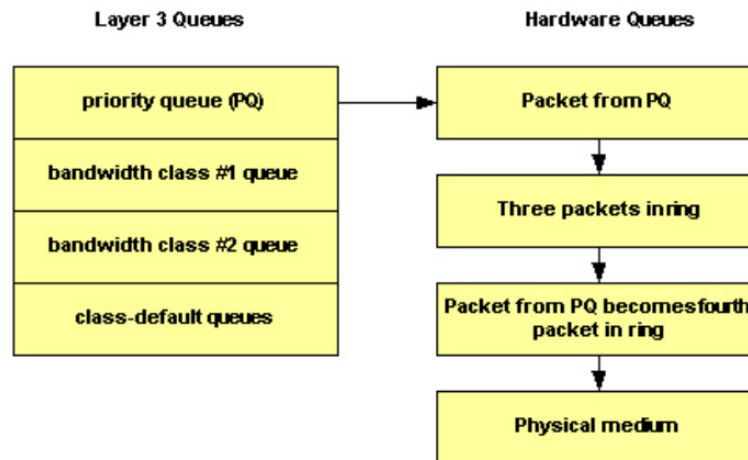
Queue	Location	Queueing Methods	Service Policies Apply	Command to Tune
Hardware queue or transmit ring	Port adapter or network module	FIFO only	No	tx-ring-limit
Layer 3 queue	Layer 3 processor system or interface buffers	Flow-based WFQ, CBWFQ, LLQ	Yes	Varies with queueing method. Use the queue-limit command with a bandwidth class.

Fonte: (Cisco Systems, 2008)

Para melhor compreensão do comportamento de um pacote na fila do roteador, será explicada a lógica da Figura 4.2.

Considere a fila com os quatro pacotes na camada 3. A fila de *hardware* foi configurada para manter até quatro pacotes, três deles já estão prontos para serem transmitidos, mas chega o quarto com prioridade de encaminhamento. Nesse caso, o mecanismo LLQ (*Low Latency Queueing*) usado como exemplo de controle, simplesmente retira o pacote recém chegado

Figura 4.3: Fluxo do pacote marcado com prioridade



Fonte: (Cisco Systems, 2008)

com prioridade, colocando-o no início da fila para ser transmitido.

Uma vez que há um gerenciamento da largura de banda garantida para fluxos de dados, caso essa reserva não tenha sido totalmente configurada para uso exclusivo de pacotes prioritários, a banda restante será distribuída da seguinte maneira: a largura de banda restante será compartilhada por todas as classes e a classe padrão (Cisco Systems, 2008, p. 6).

Para exemplificar a forma como o mecanismo atua, considere um *link* de 1,544 Kbps que foi configurado para reservar 30% para fluxos de voz, e 60% destinados à dados. Portanto, 463.2 Kbps foram reservados para voz, e 926.4 Kbps para os dados. Se a classe padrão não usar os 10% restantes, o equivalente a esse percentual, ou seja, 154,4 Kbps, serão compartilhados na mesma proporção a que foram configurados: 30:60 ou 46.32 Kbps: 92.64 Kbps.

4.2.1 A escolha do software para simulação de vídeo sob demanda

Há muitas alternativas de *softwares freeware* disponíveis no mercado: PleX, XSplit, VideoLAN, e etc. A escolha pela solução da VideoLAN, versão 2.2.2 Weathermax, revisão 2.2.2-0-g6259d80 (Linux (CentOS 7 e Fedora 22) e Windows), se deu pela facilidade de uso, e pela existência de versões para as plataformas MacOS, Android e iOS, além de ser bastante utilizado em laboratório de testes em trabalhos acadêmicos.

Os principais protocolos existentes no VLC são:

- *HyperText Transfer Protocol* (HTTP),
- *Real-time Transport Protocol* (RTP) / *Moving Picture Experts Group* (MPEG),
- *Real-time Transport Streaming Protocol* (RTSP) e
- *User Datagram Protocol* (UDP) (*legacy*).

A próxima seção detalhará o processo de escolha do protocolo para a avaliação da experiência do usuário.

4.3 Caminho percorrido para os testes

Durante a fase de ajustes no *hardware* e *software* do ambiente proposto, aconteceram situações que provocaram incertezas quanto à validação do cenário. Como o PC do fabricante LOGIN apresentou incompatibilidade com a placa NetFPGA, optou-se usá-lo como servidor de vídeo, cuja versão do *software* VLC instalada foi 2.1.6, e o Sistema Operacional instalado foi o Linux Mint 17.3².

Com o decorrer dos testes, este também não se mostrou confiável. O ponto desfavorável no *hardware* desse fabricante é a sua instabilidade com o ambiente Linux. Sem qualquer explicação lógica, o *software* VLC deixava de ser reproduzido no cliente durante a determinação do *baseline* para uso do protocolo mais apropriado aos testes. O software era encerrado abruptamente ao tentar reproduzir o filme via HTTP; já com o protocolo RTSP, este passou a reproduzir com baixa qualidade de vídeo, condições essas que não foram reproduzidas outrora, cuja velocidade de transmissão estava a 1Gbps durante todos os momentos de testes.

Uma possível explicação para esse padrão de funcionamento foi a detecção de excessivo consumo de CPU no *desktop* que foi configurado para ser o servidor de *streaming*.

Nos primeiros ensaios, a recomendação de algumas comunidades na Internet que usavam o VLC como servidor de *streaming* quanto às configurações de transmissão foram:

²as distribuições Linux possuem pacotes de *softwares* homologados, e essa versão do SO só disponibilizou a que foi determinada como estável, apesar da empresa VIDEOLAN já possuir atualizações da versão do VLC.

- desabilitar a opção “ativar transcodificação”;
- desabilitar a opção “fluxo de todos os fluxos elementares”.

A razão para tal configuração tinha como consequência um elevado consumo de processamento no servidor de vídeo, a ponto de ocupar em 100% o consumo nos quatro núcleos da CPU do PC da marca LOGIN, impactando diretamente na transmissão do *streaming*. Porém, em determinado momento, ao deixar a primeira opção ativada, o consumo médio de CPU ficou abaixo de 40%, mas voltando a condição anterior após vários dias de testes.

Como não houve atualização de *kernel* ou do VLC que pudesse provocar tal mudança no comportamento do serviço de *streaming*, não foi possível chegar a alguma conclusão que pudesse determinar se há, ou não, qualquer incompatibilidade da arquitetura dos computadores desse fabricante, porque, ao instalar a versão do Windows 10 Pro nesse *hardware*, houve melhora na estabilidade durante a fase de testes, condizentes com os resultados que se esperava obter em termos de funcionalidade do VLC. Um forte indicativo é que o sitio desse fabricante não disponibilizou *drivers* para distribuições Linux, podendo ser essa uma das causas dessa instabilidade.

4.3.1 Determinação da Linha de Base

Foram criados três cenários para a definição da linha de base. No primeiro, apenas os roteadores Cisco, um servidor de *streaming* de vídeo e uma estação-cliente. O segundo cenário, somente a placa NetFPGA, o servidor de *streaming* e a estação-cliente.

Durante esses testes, tendo o protocolo RTSP sobre UDP como padrão para a transmissão dos fluxos, foram transmitidos dois filmes: um em baixa resolução (O Dólar Furado, a 640 x 352 ppi) e outro em alta resolução (Insurgente, a 1.920 x 800 ppi). Como o filme em baixa resolução teve um pequeno impacto, foi considerado para a avaliação somente o filme em alta resolução. Em todos os cenários, apenas o *link* com a estação-cliente é wifi, usando o protocolo IEEE 802.11n. Os demais enlaces usam cabeamento metálico Cat.6.

Para determinar se o filme seria executado no cliente o mais fiel possível da versão original, foi feita uma conexão direta entre o servidor VLC e o cliente, utilizando o protocolo RTSP por possuir as características necessárias para transmissão de fluxos multimídia. Mas, para surpresa, este não se mostrou com qualidade aceitável. Visualmente, era percebido

pequenas paralisações durante a reprodução do filme, que era relatado nas estatísticas do próprio *software* VLC do cliente. A Figura 4.4 foi obtida de um cliente executando o filme via UDP.

Figura 4.4: Estatísticas do *software* VLC. Cliente.

Mídia atual / estatísticas de fluxo	
▼ Áudio	
Decodificado	6776 blocos
Reproduzido(s)	6771 memórias
perdido(s)	0 memórias
▼ Vídeo	
Decodificado	2178 blocos
Exibido	2254 quadros
perdido(s)	85 quadros
▼ Entrada/Leitura	
Tamanho de dados de mídia	0 KiB
▶ Taxa de bits de entrada	0 kb/s
Tamanho do dado descombinado	52013 KiB
Taxa de bits do conteúdo	1565 kb/s
Descartados (corrompidos)	0
Descartado (descontinuado)	0
▼ Saído/Escreito/Enviado	
Enviado(s)	0 pacotes
Enviado(s)	0 KiB
Taxa de bits do fluxo de envio	0 kb/s

Fonte: *software* reprodutor multimídia VLC

Ao analisar o relatório gerado pelo *software* VLC no lado cliente, foram apresentadas as seguintes mensagens de erro:

1. *MultiFramedRTPSource::doGetNextFrame1(): The total received frame size exceeds the client's buffer size (100000). 31964 bytes of trailing data will be dropped!*
2. *[h264 @ 0x7f8424015900] reference picture missing during reorder;*
3. *[h264 @ 0x7f8424015900] Missing reference picture, default is 2;*
4. *[h264 @ 0x7f8424015900] decode_slice_header error;*
5. *[h264 @ 0x7f8424015900] Frame num change from 2 to 0;*

6. [h264 @ 0x7f841c1341a0] Missing reference picture, default is 65566.

O erro do item 2 indica que o quadro I (Intracodificado) do padrão MPEG (*Moving Picture Experts Group*) está se perdendo. Esse quadro tem a função de permitir que um usuário qualquer consiga assistir a um filme no momento que desejar, e essas perdas fazem com que a reprodução do *streaming* fique a todo momento se remontando, provocando uma experiência muito ruim ao usuário. Esse erro ocorreu durante todo o tempo da transmissão.

O erro do item 3 ocorreu logo após no início da reprodução, provocado pelo erro anterior, mas que, provavelmente, desencadeou a sequencia de reordenação do item 1 e demais erros.

Foi importante observar que, durante a investigação das causas dos problemas identificados, houve necessidade de maior investigação sobre o protocolo MPEG. Este, segundo a especificação, não determina que o fluxo na origem seja idêntico no destino; há uma tolerância a perdas, cuja fundamentação está na forma como o olho humano percebe imagens em movimento. A nossa retina retém uma imagem por alguns milissegundos antes de ser substituída por outra, ou mesmo esquecida. À taxa de reprodução de 50 imagens por segundo, o olho não é capaz de perceber essa variação. Por esse motivo, aceitam-se perdas no receptor, desde que essas não sejam grandes o suficiente a ponto de se tornarem perceptíveis.

Ajustes na velocidade de transmissão das interfaces de rede

Esse processo se fez necessário para que não precisasse adquirir interfaces de rede com velocidade de transmissão específica. Para tal ajuste, alguns parâmetros foram utilizados para adequação às exigências dos testes. No CentOS, os utilitários utilizados foram:

- `# ethtool -s ``nome da interface`` 100 duplex full autoneg off, ou`
- `# mii-tool ``nome da interface`` --force=100baseTx-FD`

onde “#” representa o comando em modo superusuário, `ethtool` para alterar a velocidade e o restante da linha os parâmetros que determinam o modo de transmissão. Esse comando coloca a interface de rede no modo 100 Mbps FULL DUPLEX. Para alterar à 10 Mbps, basta mudar o valor do parâmetro de 100 para 10.

Se usar o utilitário `mii-tool` e quiser mudar para 10 Mbps, o comando será:

- # mii-tool ``nome da interface'' - -force=10baseT-FD.

Para restaurar as configurações de fábrica, execute:

- # mii-tool ``nome da interface'' -R,
- # ethtool -s ``nome da interface'' 1000 duplex full autoneg on.

Como as interfaces de rede utilizadas são, por padrão, *Gigabit Ethernet*, não foi necessário realizar quaisquer configurações para essa velocidade de transmissão.

Na subseção a seguir, serão apresentados os testes dos protocolos e suas estatísticas nas taxas *Gigabit Ethernet*, *Fast Ethernet* e *Ethernet*.

4.4 Teste dos protocolos

Foi observado que há diferenças significativas quanto ao comportamento dos protocolos. Os testes iniciais foram executados usando o protocolo RTSP, voltado para fluxos multimídia. Porém, este apresentou um comportamento diferente quando a taxa de transmissão nas interfaces de rede dos dispositivos foram alteradas de 1 Gbps (*Gigabit Ethernet*) para 100 Mbps (*Fast Ethernet*) e 10 Mbps (*Ethernet*).

Um ponto considerado como possível causa de problemas entre transmissor e receptor foi o tamanho de *buffer*. As mensagens de erros apresentadas na subseção anterior, todas ocorridas ao usar o protocolo UDP, indicavam que, ao ajustar a quantidade de memória com valores suficientes para receber os fluxos multimídia, resolveriam tal limitação. Mas, ao consultar o *buffer* UDP disponibilizado pelo S.O. no ambiente Linux, foram obtidas as seguintes informações:

- \$ cat /proc/sys/net/core/rmem_default = 212992 bytes;
- \$ cat /proc/sys/net/core/wmem_default = 212992 bytes;
- \$ cat /proc/sys/net/core/rmem_max = 212992 bytes;
- \$ cat /proc/sys/net/core/wmem_max = 212992 bytes;

onde *rmem* e *wmem* são as variáveis *read memory* e *write memory*. Outro parâmetro analisado foi específico do IPv4 para o protocolo UDP, conforme a seguir:

- `$ cat /proc/sys/net/ipv4/udp_mem = 127173 169564 254346 bytes;`

onde os três valores representam, respectivamente, mínimo, *default* e máximo.

Esses valores são definidos conforme a memória RAM disponível. No caso, o *desktop* tinha 4GB de RAM, e o servidor, 8GB de RAM.

Sabendo-se que a MTU (*Maximun Transmission Unit*) da camada de enlace é de 1.500 *bytes*, e que o protocolo UDP não negocia o tamanho da janela de transmissão por ser um protocolo não orientado a conexão, pode-se concluir que os valores definidos pelo S.O. são suficientes para acomodar as rajadas dos fluxos transmitidos pelo servidor de *streaming*.

As configurações do VLC no cliente assumiram um *cache* que permitisse “bufferizar” o fluxo por um segundo, quanto ao lado servidor, este foi mantido o padrão de 0,3 segundos. Houve necessidade de encerrar e executar o *software* VLC no servidor porque apresentava os erros “**cannot add new streaming**”, “**cannot create packetizer output (h264)**”, consequentemente o cliente não conseguia reproduzir o *streaming*.

Para que fluxos *multicast* fossem reproduzidos, houve necessidade de se criar rotas tanto no lado servidor, quanto no lado cliente. O comando usado no S.O. Linux foi:

- `# route add -net 224.0.0.0 netmask 240.0.0.0 dev [nome da interface]`

Para S.O Windows, deve-se digitar:

- `route add 224.0.0.0 mask 240.0.0.0 ``IP da interface`` metric 2`

O processo de transmissão dependerá do protocolo a ser utilizado para criação do fluxo multimídia.

Foi definido um intervalo de tempo igual a 5 minutos para cada protocolo, e cada um dos testes alterando as velocidades de transmissão nas interfaces de rede. Os testes foram executados conforme os procedimentos a seguir.

1. Iniciar o VLC no servidor;
2. Criar um fluxo multimídia;
3. Adicionar o protocolo a ser usado;
4. Definir uma porta para acesso à aplicação (recomenda-se não alterar o padrão fornecido pelo *software*);
5. Desmarcar a opção “ativar transcodificação”;
6. Ativar (ou não, dependerá do protocolo selecionado) a opção “fluxo de todos os fluxos elementares”;
7. Pressionar o botão “fluxo”.

No lado cliente, o procedimento difere um pouco, mas deve estar de acordo com o padrão especificado no lado do servidor:

1. abrir fluxo de rede e especificar o protocolo, endereço IP do servidor e a porta da aplicação, de acordo com o exemplo exibido na janela de configuração;
2. pressionar o botão “play”.

Para que se consiga obter resultados mais próximos do ambiente real, se fez necessário ajustar as taxas de transmissão da interface de rede do servidor VLC.

A seguir, se apresentam os comportamentos dos fluxos multimídia com relação ao protocolo utilizado e à taxa de transmissão nas interfaces de rede a 1 Gbps, 100 Mbps e 10 Mbps.

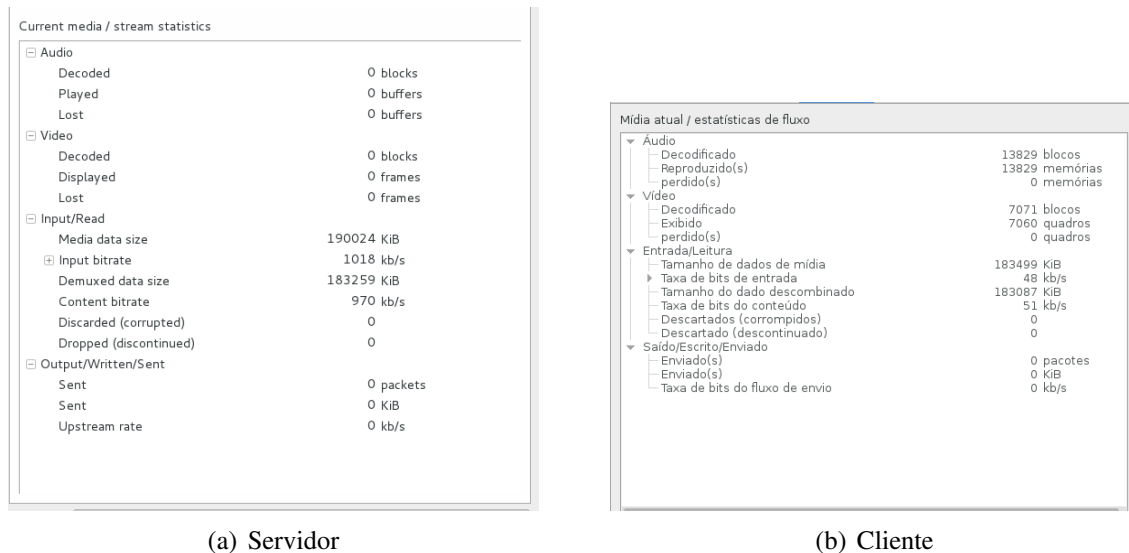
Protocolo HTTP

Como os protocolos RTSP, RTP e UDP, nos testes iniciais, tiveram desempenho aquém do esperado à taxa *Gigabit*, optou-se por homologar o protocolo HTTP sobre TCP para a resolução à 1.920 x 800 ppi e a 1.280 x 720 ppi. As causas do baixo desempenho do protocolo de transporte UDP foram apresentadas na subseção 4.3.1.

O protocolo HTTP, quando transmitido à taxa de 1 Gbps, não apresentou perdas dos quadros de áudio e vídeo, a experiência do usuário foi excelente, sem quaisquer interrupções

ou remontagem da imagem reproduzida. Durante os 5 minutos de reprodução, as estatísticas do VLC no lado cliente demonstram esse comportamento (Figura 4.5).

Figura 4.5: Estatísticas do *software* VLC a 1 Gbps



Fonte: *software* reprodutor multimídia VLC

Ao alterar a taxa de transmissão da interface de rede no servidor para 100 Mbps, o cenário era:

- Servidor Linux CentOS 7;
- Cliente Linux Fedora 22.

Nessa condição, o comportamento no cliente não foi reproduzido da mesma forma ao repetir o experimento. Nos primeiros seis segundos iniciais, o filme deixava de ser reproduzido, mesmo havendo transmissão dos fluxos e estes sendo recebidos no cliente. A mensagem era de que o *buffer* estava sendo reconfigurado. Mesmo assim, o filme não mais era reproduzido. Ao fazer o ajuste manual do “cache da rede” no *software* VLC, ainda assim o problema se repetiu.

Porém, ao instalar o VLC no *desktop* da fabricante LOGIN, cujo S.O. instalado é Windows 10 Pro, este comportou-se como se estivesse a 1 Gbps, ou seja, mesmo a 100 Mbps, o filme foi reproduzido sem travamentos, tendo o mesmo cliente consumindo o *streaming* produzido pelo servidor.

A seção a seguir apresentou a metodologia de medição da experiência do usuário com base nas recomendações do Comitê Internacional para Padronização das Telecomunicações.

4.5 Métricas da Qualidade de Experiência

A recomendação do ITU-T para mensurar a qualidade da experiência do usuário é através da média de um placar de opiniões (MOS - *Mean Opinion Score*), cujas perguntas são objetivas, mas a avaliação do usuário é subjetiva. No experimento, foi avaliado um filme transmitido em alta resolução e mensurada a experiência em momentos específicos, cuja meta foi observar se as variações nas condições da rede, ao injetar um tráfego aleatório, foram compensadas com ajustes dos parâmetros de NQoS e AQoS. Foi entregue um formulário com as seguintes perguntas quanto à SATISFAÇÃO:

- Excelente (acima de 89%)
- Bom (entre 70% e 89%)
- Regular (entre 50% e 69%)
- Ruim (entre 30% e 49%)
- Impossível avaliar (abaixo de 30%)

A classificação por percentual servirá de referência para indicar o grau de satisfação em relação ao conteúdo assistido.

4.5.1 Configuração dos dispositivos dos observadores e procedimentos de avaliação

Os observadores utilizaram seus próprios equipamentos, todos os dispositivos foram *notebooks*. A intenção foi poder instalar agentes SNMP para coleta dos contadores de CPU, memória e interface de rede. Todos os dispositivos tinham como sistema operacional versões do Windows.

Para poder controlar ao máximo as variáveis do ambiente de teste, foi determinado o seguinte padrão de configuração nos dispositivos dos observadores:

- Bateria do dispositivo totalmente carregada;
- Modo de configuração de uso da bateria equilibrado;

- Instalação do *software* VLC;
- Instalação/configuração do agente SNMP.

Não houve interferência quanto à quantidade de processos em execução, nem a velocidade de processamento, ou mesmo quantidade de memória RAM disponível. Porém, todas essas informações foram coletadas e inseridas nos formulários de avaliação dos observadores.

A estratégia de testes seguiu a seguinte ordem, realizada em duas etapas:

- Reprodução do filme INSURGENTE, cujas resoluções foram 1.920 x 800 ppi e 1.280 x 720 ppi;
- Cada observador devidamente posicionado;
- Após cada reprodução, os observadores marcaram uma das opções do formulário;

Na etapa seguinte, a única alteração quanto a ordem de execução foi apenas em relação aos observadores trocarem suas posições, permanecendo os dispositivos em seus locais de origem.

A bateria dos dispositivos de alguns dos participantes estava sem carga. Essa restrição causou uma pequena mudança na estratégia de testes, obrigando a todos usarem seus equipamentos energizados na tomada. Não houve prejuízo no resultado.

Cada observador avaliou o vídeo obedecendo a seguinte estratégia:

- A primeira transmissão sem política de QoS aplicada nas interfaces dos roteadores e sem injeção de tráfego na rede, através do *software* *iPerf*;
- A segunda transmissão sem política de QoS, com injeção de tráfego na rede utilizando o protocolo TCP;
- A terceira transmissão com a política de QoS aplicada nas interfaces de rede dos roteadores, sem injeção de tráfego na rede.
- A quarta transmissão com a política de QoS aplicada nas interfaces de rede dos roteadores, com injeção de tráfego na rede.

De posse dos resultados estatísticos, foi feito um cruzamento com as variações do ambiente controlado, comparando-as aos parâmetros de QoS quanto ao fluxo de dados no canal de comunicação.

4.6 Considerações finais do Capítulo

A avaliação inicial demonstrou que o protocolo UDP, apesar de ser considerado adequado para *streaming* multimídia, não apresentou bom desempenho tanto a *Gigabit*, quanto em modo *FastEthernet*. Porém, não é conclusiva, porque o padrão MPEG permite implementações considerando perdas, e o *software* VLC, por ser fechado e não ser parte do escopo desse trabalho, não foi avaliado quanto a essas perdas.

Foi percebido que há necessidade de ajustes no QoS, uma vez que os roteadores Cisco marcam pacotes sem classificação como *Best Effort*. Portanto, mesmo não havendo outros fluxos, estes entrarão em uma fila antes de serem encaminhados ao seu destino, provocando certo atraso à transmissão.

Com relação aos procedimentos para os testes, o objetivo foi centralizar os dados dos dispositivos dos usuários na ferramenta de análise de rede **Cacti** para a geração dos gráficos de desempenho.

Capítulo 5

Avaliação das condições perceptuais do usuário em transmissão de fluxo de dados

Este capítulo tem por objetivo apresentar a metodologia utilizada nos testes de avaliação da percepção do usuário e topologia da rede, considerando como fatores de influência nessa experiência:

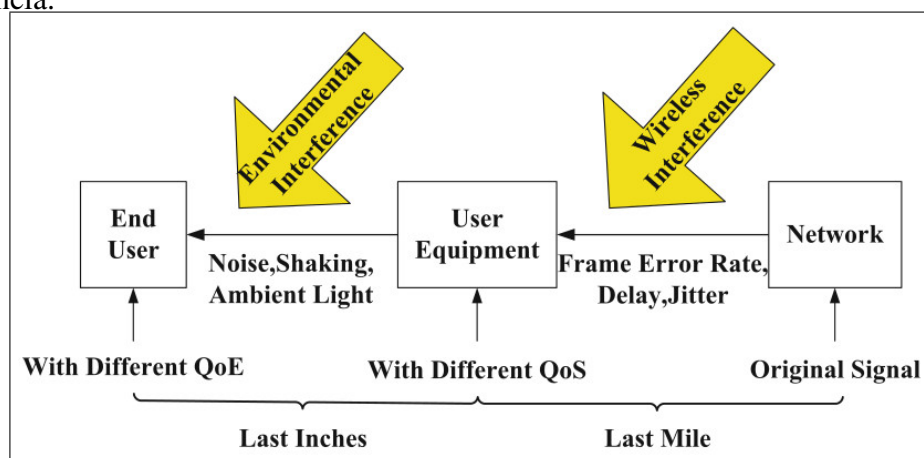
- Os parâmetros de QoS da rede (*Network QoS*);
- Os parâmetros de QoS da Aplicação (*Application QoS*);
- Qualidade do sinal recebida pelo Equipamento do Usuário (EU);
- As características dos dispositivos dos usuários.

No artigo sobre modelos de correlação QoS/QoE em serviços multimídia (ALRESHODI; WOODS, 2013), avaliou oito modelos, dos quais dois são objetos desse trabalho – avaliação de QoE baseado na arbitragem de recursos da rede (SILLER; WOODS, 2003), e considerando fatores ambientais e equipamentos dos usuários (HAN et al., 2012). Essa proposta de trabalho correlacionando QoS/QoE teve um *paper* (JACAUNA; MORENO; SALGUEIRO, 2016) aprovado no AICT (*Advanced International Conference on Telecommunications*).

De acordo com o modelo proposto na Figura 5.1 (HAN et al., 2012), eles concluíram que a experiência do usuário é determinada por todos os três fatores a seguir:

- Equipamento do Usuário (EU);

Figura 5.1: Modelo de QoE considerando equipamentos e ambiente como fatores de interferência.



Fonte: (HAN et al., 2012)

- Qualidade do sinal recebida pelo EU, e;
- Interferência do ambiente no EU.

Porém, não leva em consideração as características dos parâmetros de QoS, por entenderem que são os fatores citados que determinarão, de fato, uma melhor, ou pior, experiência do usuário durante a transmissão de um serviço de *streaming*. Eles não consideraram, também, a característica do equipamento.

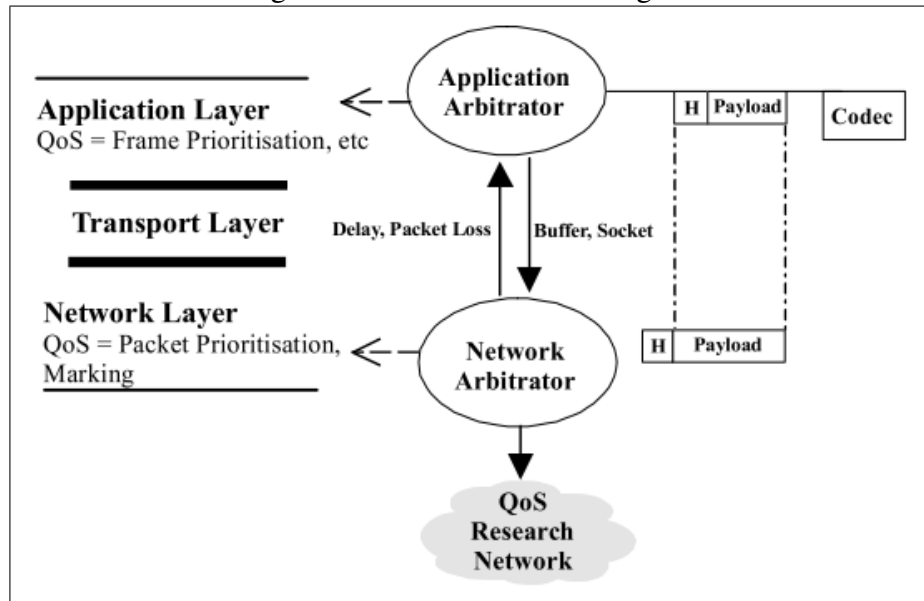
Com relação ao modelo proposto por Siller *et.al.* (2003), Figura 5.2, foi desenvolvido um *Framework* baseado na hipótese de que, se os serviços nas camadas de rede e aplicação foram arbitrados, uma melhor experiência do usuário poderá ser obtida.

Como no modelo proposto por Alreshoodi *et.al.*, não foi considerado a característica do dispositivo final que o usuário estará usando.

Nos primeiros testes para a definição da linha de base, já ficou demonstrado que todos os fatores que foram levantados influenciam diretamente nas condições perceptuais do usuário, incluindo a configuração de *hardware* do servidor de *streaming*.

Na próxima seção, será apresentado a execução dos testes com as características de *hardware* e *software*.

Figura 5.2: Camada de Arbitragem.



Fonte: (SILLER; WOODS, 2003)

5.1 Execução do experimento

Esta seção descreve o processo de realização do experimento em dois ambientes distintos: nas instalações do IFS, campus São Cristóvão, subseção 5.1.1, e na UFS, subseção 5.1.2. O objetivo de realizar em ambientes separados foi para determinar qual resolução do vídeo seria mais adequada para demonstrar o comportamento da rede, aplicação, equipamentos e usuários envolvidos.

O cenário geral foi praticamente o mesmo, com mudança no dispositivo que serviu de infraestrutura para os usuários finais, como será descrito nas subseções a seguir.

Para fins de avaliação da experiência do usuário, o cenário com roteadores Cisco foi escolhido para efeito de comparação tanto dos usuários, quanto dos dispositivos, adicionado um comparativo dos protocolos HTTP e RTSP. Concluiu-se que o ambiente com a placa NetFPGA nada influenciaria por ainda não ter sido desenvolvido um projeto que permita configurar parâmetros de QoS em uma rede de computadores.

5.1.1 Testes no IFS

A Figura 5.3 apresenta o espaço na qual foi instalado, configurado e realizados todos os testes de QoS e QoE. Os dois roteadores Cisco e o equipamento de multipropósito TP-LINK aparecem na Figura 5.4, ao passo que as interfaces de rede da placa NetFPGA é exibida através da Figura 5.5. Por fim, os clientes Windows que serviram de base para o experimento, ambos usando conectividade *wireless* padrão 802.11n (Figura 5.6).

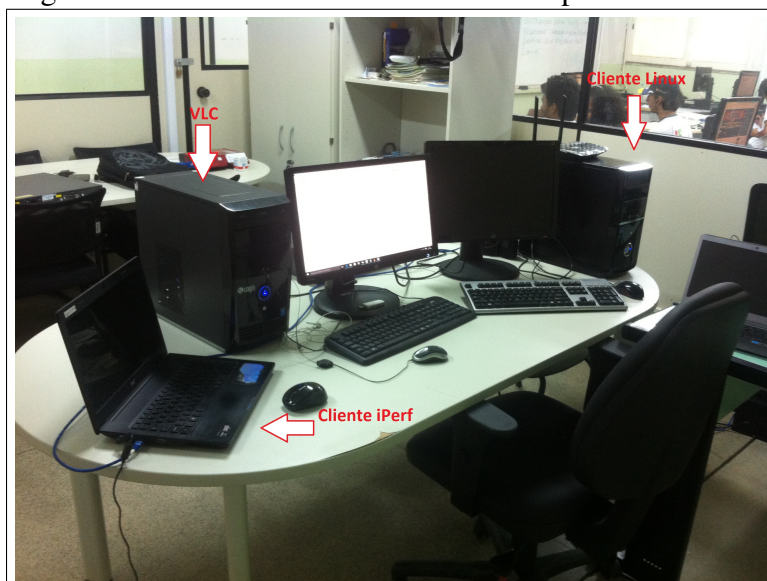
O uso do dispositivo de multipropósito TP-LINK em lugar do DLINK-DIR 610 se deu porque este possui interfaces *Fast Ethernet*, incompatíveis com as interfaces de rede da placa NetFPGA e, como o cenário foi subdividido em dois, cada qual com suas características únicas, adotou-se o TP-LINK como padrão para minimizar a quantidade de variáveis no ambiente a ser controlado.

Os dispositivos que representaram os usuários finais possuíam as seguintes especificações técnicas:

- Ultrabook Samsung
 - Windows 10 *Home Single Language* de 64 bits;
 - Processador AMD A4-4335M APU 1,9 GHz, com interface gráfica Radeon HD;
 - Memória com 6,0 GB RAM;
 - Disco Rígido com capacidade para armazenamento de 500 GB.
- HP Pro One 400 G1 AiO
 - Windows 7 Professional de 64 bits;
 - Processador Intel Core i5 2,5 GHz;
 - Memória com 4 GB de RAM;
 - Disco Rígido com capacidade para armazenamento de 500 GB.

Durante os experimentos iniciais, os resultados foram ruins, sem contar com a dificuldade na adesão de participantes que se predispusessem a avaliar o experimento. Outro fator negativo foi o dispositivo móvel de um dos participantes só funcionar na tomada. Isso inviabilizou a ideia inicial de fazer com que cada observador assistisse ao trecho do vídeo em cada um dos dispositivos existentes.

Figura 5.3: Foto. Servidor VLC e PC com placa NetFPGA.



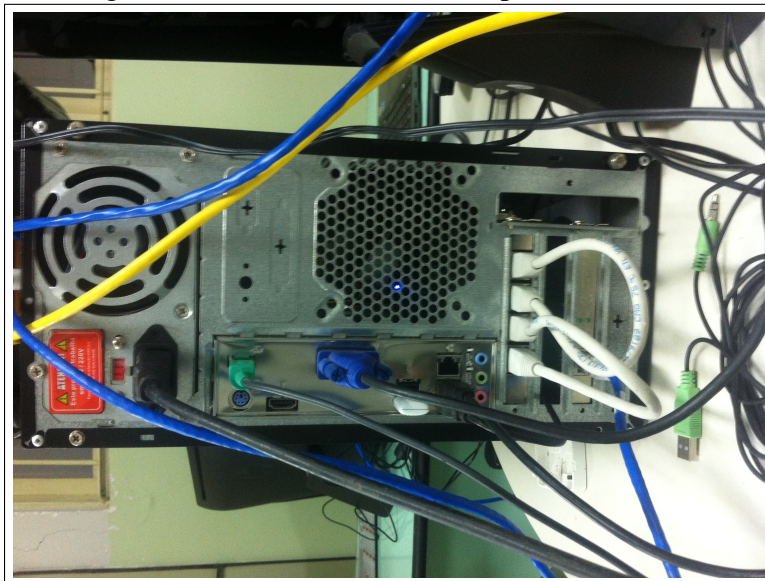
Fonte: Elaborado pelo autor.

Figura 5.4: Roteadores Cisco e dispositivo multipropósito TP-LINK.



Fonte: Elaborado pelo autor.

Figura 5.5: Interfaces de rede da placa NetFPGA



Fonte: Elaborado pelo autor.

Figura 5.6: Interfaces de rede da placa NetFPGA

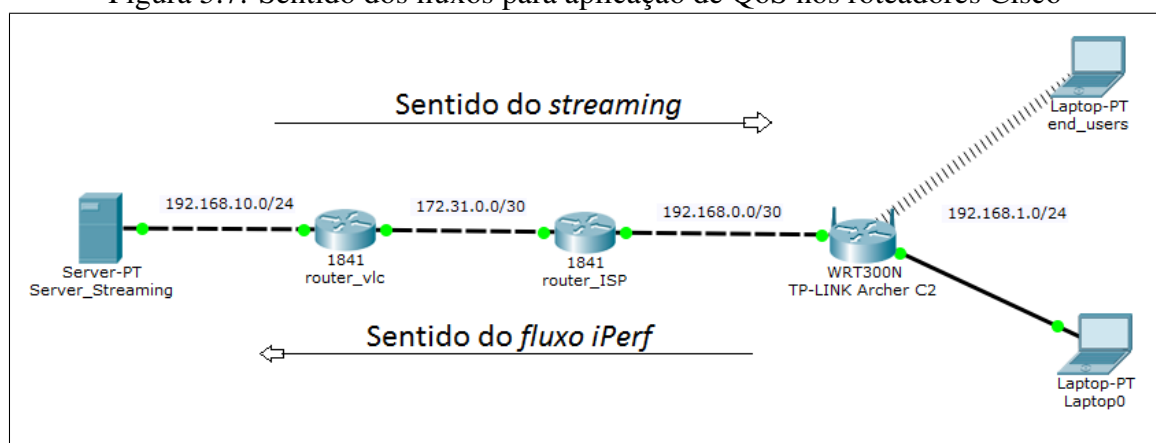


Fonte: Elaborado pelo autor.

Diante dessas limitações, optou-se por trabalhar com apenas dois dispositivos com interfaces *wireless*, executando o filme INSURGENTE durante os primeiros vinte e dois minutos, cujas variações de cenas em modo *FULL HD* atinge pico de até 16 Mbps. Porém, ao iniciar o filme, a taxa de *bits* do conteúdo chega próximo de 30 Mbps. Como são duas estações reproduzindo o mesmo filme, dobra-se esse valor, mas há algo que deve ser levado em consideração: a probabilidade de duas estações iniciar o filme no mesmo instante é muito baixa, ou quase zero, porque o servidor estará atendendo a requisição daquele que pediu primeiro. A consequência disso é que tempos diferentes sobrepõem-se e, a depender do início da execução do filme na estação seguinte, poderá haver um tempo maior de utilização da largura de banda disponível em cenas com maior movimentação. Essa condição fez com que novos ajustes de QoS fossem feitos para adequar às características de reprodução do filme.

O cenário com os roteadores Cisco é apresentado na Figura 5.7. Para melhor compreensão de como as regras de QoS devem ser aplicadas às interfaces dos roteadores, essa figura indica o sentido entre origem e destino. Apesar de usar o servidor VLC tanto para *streaming*, quanto para fluxos do *iPerf*, a bidirecionalidade para transmissão e recepção de dados na interface desse servidor não impactou em seu desempenho, porque ambos os fluxos são caracterizados como *half duplex*. Isso não quer dizer que os fluxos sejam trafegados apenas nesse sentido. A ferramenta *iPerf* permite que haja bidirecionalidade na comunicação, basta que se altere um parâmetro na ferramenta.

Figura 5.7: Sentido dos fluxos para aplicação de QoS nos roteadores Cisco



Fonte: Elaborado pelo autor.

As próximas subseções demonstraram os gráficos de desempenho do cenário utilizando

roteadores Cisco. Para que haja um padrão a ser seguido, todas as execuções tiveram os *softwares* VLC finalizados e reiniciados nos três dispositivos – servidor e as duas estações de trabalho – para minimizar, ao máximo, inconsistências nos testes. Todos iniciaram no tempo zero e foram finalizados após atingir 22” de reprodução.

5.1.2 Testes na UFS

Nesse ambiente, foram recrutados alunos do curso da área de Computação, cada aluno com seu respectivo dispositivo, conforme tabela 5.1.

Tabela 5.1: Características dos dispositivos utilizados no experimento

Dispositivo	CPU (clock)	Display	RAM	GPU
E1 – Samsung Galaxy Young	1,0 GHz	480x320	768 MB	Não
E2 – Samsung Galaxy S4 mini	1,7 GHz	540x960	1536 MB	Sim
E3 – Notebook DELL	2,2 GHz	1366x768	4096 MB	Sim
E4 – Asus Zenfone 2	2,3 GHz	1920x1080	4096 MB	Sim
E5 – Moto G2	1,2 GHz	1280x720	1024 MB	Sim
E6 – Ultrabook Samsung	1,9 GHz	1366x768	6144 MB	Sim

Fonte: Elaborado pelo autor.

Para a avaliação da percepção do usuário, foi utilizado um vídeo com duração de 1 minuto e 2 segundos à resolução de 720i (1.280 x 720 ppi), utilizando os protocolos HTTP e RTSP. O motivo foi a dificuldade em obter dispositivos que executassem vídeos na resolução de 1080 ppi, além de limitar o número de participantes devido às rajadas em determinadas cenas, já detectadas no experimento dos testes no IFS.

Porém, apesar de o protocolo UDP já ter sido caracterizado como não apropriado para testes em vídeos com resolução FULL HD, para fins de avaliação e comparação de desempenho entre protocolos, na resolução de 720i, foi possível transmitir as rajadas em cenas com intensas mudanças de cenas por atingirem, no máximo, 4 Mbps aproximados.

Como os testes em FULL HD também foram realizados nesse ambiente, com o 6 participantes, a mesma experiência obtida no IFS se repetiu, não proporcionando uma avaliação mais adequada à proposta do experimento. Assim, foi possível concluir que, baixando a resolução do vídeo, satisfaria as necessidades de obter informações que dessem suporte a argumentação na escolha dos protocolos, e resoluções do vídeo a serem avaliados pelos

usuários.

5.2 Cenário com roteadores Cisco nas instalações do IFS

Foram realizados quatro rodadas de vinte e dois minutos, seguindo a seguinte ordem de execução:

1. Sem QoS aplicado nas interfaces de rede dos roteadores e sem carga do *iPerf*;
2. Sem QoS aplicado nas interfaces de rede dos roteadores, mas com carga do *iPerf*;
3. Com QoS aplicado nas interfaces de rede dos roteadores e sem carga do *iPerf*;
4. Com QoS aplicado nas interfaces de rede dos roteadores e com carga do *iPerf*.

Pode parecer estranho realizar testes com QoS sem carga, mas o objetivo foi identificar se as regras poderiam, ou não, influenciarem no desempenho da rede.

5.2.1 Sem QoS nas Interfaces de rede dos roteadores, sem carga do *iPerf*

Ao longo do período estabelecido para os testes, não foi identificado quaisquer alterações na qualidade da imagem, que pode ser comprovado através da Figuras 5.8 e 5.9, que representam, respectivamente, um Ultrabook Samsung e um HP Pro One 400 G1 AiO.

Houve necessidade de adicionar um *plugin* para medição em tempo real na ferramenta Cacti, para dar maior veracidade na coleta dos dados das interfaces de rede das estações. Mas, ainda assim, o intervalo de coleta pode mascarar algum intervalo relevante, uma vez que o mínimo de configuração permitido são cinco segundos por amostragem. Contudo, foi editado o arquivo ```setup.php``` e adicionado o parâmetro no *array* para permitir leituras a cada segundo.

Foi observado que o filme, na transmissão dos primeiros 20" iniciais, atingiu uma taxa máxima de até 40 Mbps por estação, totalizando na interface de rede do servidor de *streaming* algo em torno de 80Mbps (Figura 5.10) com duas estações recebendo os fluxos. Esses valores podem estar alterados em função de alguma carga adjacente, porque ao observar o

Figura 5.8: Estatísticas VLC do Ultrabook

▼ Áudio	
Decodificado	61350 blocos
Reproduzido(s)	61350 memórias
perdido(s)	36 memórias
▼ Vídeo	
Decodificado	31377 blocos
Exibido	31351 quadros
perdido(s)	0 quadros
▼ Entrada/Leitura	
Tamanho de dados de mídia	609035 KiB
> Taxa de bits de entrada	213 kb/s
Tamanho do dado descombinado	607383 KiB
Taxa de bits do conteúdo	230 kb/s
Descartados (corrompidos)	0
Descartado (descontinuado)	0
▼ Saído/Escreito/Enviado	
Enviado(s)	0 pacotes
Enviado(s)	0 KiB
Taxa de bits do fluxo de envio	0 kb/s

Fonte: *software* reprodutor multimídia VLC

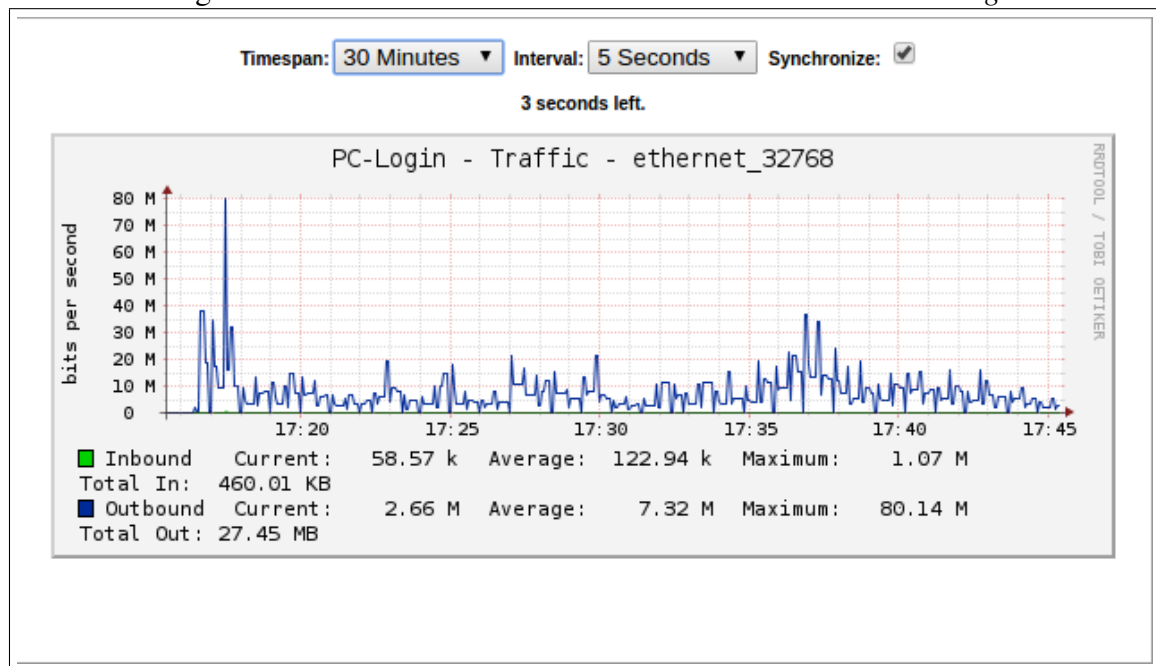
Figura 5.9: Estatísticas VLC do HP Pro One 400 G1 AiO

▲ Áudio	
Decodificado	61662 blocos
Reproduzido(s)	61662 memórias
perdido(s)	0 memórias
▲ Vídeo	
Decodificado	31536 blocos
Exibido	31510 quadros
perdido(s)	0 quadros
▲ Entrada/Leitura	
Tamanho de dados de mídia	614063 KiB
▷ Taxa de bits de entrada	72 kb/s
Tamanho do dado descombinado	612402 KiB
Taxa de bits do conteúdo	30 kb/s
Descartados (corrompidos)	0
Descartado (descontinuado)	0
▲ Saído/Escreito/Enviado	
Enviado(s)	0 pacotes
Enviado(s)	0 KiB
Taxa de bits do fluxo de envio	0 kb/s

Fonte: *software* reprodutor multimídia VLC

gerenciador de tarefas do Ultrabook nesse mesmo período de tempo, foi identificado um pico máximo de 28 Mbps arredondados. Esse valor permite concluir que, em uma rede cujo enlace seja de 100 Mbps, no máximo 3 estações conseguirão assistir ao filme sem quaisquer problemas de reprodução.

Figura 5.10: Estatísticas Interface de Rede Servidor de *streaming*



Fonte: *software* de monitoramento de redes Cacti

Um ponto interessante foi o fato do Ultrabook, durante esse teste, apresentar perda de quadros de áudio, logo após o início do estabelecimento da sessão, não chegando a prejudicar a reprodução do filme. Como o início dos testes foi via computador AiO da HP, o teste foi repetido com o Ultrabook iniciando primeiro os testes, mas as perdas dos quadros de áudio também aconteceram, sem quaisquer perdas nos quadros de vídeo.

5.2.2 Sem QoS nas Interfaces de rede dos roteadores, com carga do **iPerf**

Nessa etapa, a injeção de tráfego UDP iniciou após 30'' do início da execução do filme, a uma taxa de utilização simulada da banda de 60 Mbps. Apesar de um breve erro na montagem da imagem no equipamento da HP antes dos cinco minutos iniciais, o restante foi reproduzido

sem alterações, mas a perda dos quadros de áudio no Ultrabook aconteceram. Esse atraso no tráfego *iPerf* se deu em função do alto pico na transmissão do *streaming* identificado na subseção anterior.

5.2.3 Com QoS nas Interfaces de rede dos roteadores, sem carga do *iPerf*

Ao configurar o QoS do fluxo multimídia para reservar 15% da largura de banda disponível, ambas as estações só reproduziram até cinco segundos de filme. A razão para o percentual escolhido foi baseado na média consumida ao longo dos 22 minutos estabelecidos no servidor de *streaming*. A ferramenta Cacti registrou uma média de 6,74 Mbps e um pico de 9,31 Mbps. Mas, após detectar essa discrepância na obtenção nos contadores dos roteadores, após a adição do *plugin* que permite a ferramenta Cacti plotar a cada segundo, o erro na leitura foi corrigido, fazendo com que novos ajustes nos parâmetros de QoS fossem necessários. Inicialmente pensou-se em modelar o tráfego para que uma forma de onda tipo dente de serra fosse suavizada ao máximo. Mas, como a característica do filme impõe uma boa rajada durante os primeiros 20" iniciais, essa política mostrou-se ineficiente. Para ilustrar: se o experimento fosse realizado com três estações, seria necessário reservar 90% do canal de 100 Mbps só para garantir que esse início fosse reproduzido sem erro. Adicionando mais um estação, ultrapassaria o limite do canal, congelando a transmissão, provavelmente, da última estação a estabelecer a sessão com o servidor.

Como alternativa, adotou-se a política que priorizasse o pacote na fila, ao invés de garantir uma largura mínima da banda para o *streaming*.

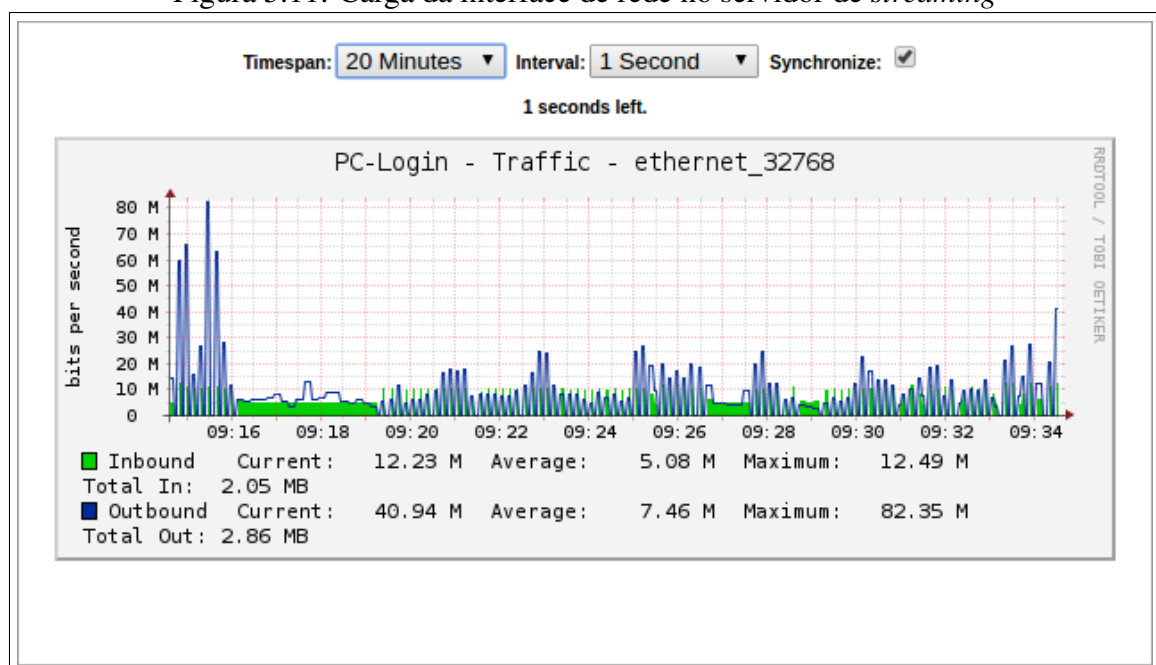
5.2.4 Com QoS nas Interfaces de rede dos roteadores, com carga do *iPerf*

Nesse experimento, foi iniciado o teste de transmissão com uma carga gerada pelo *iPerf* comprometendo 40 Mbps do canal. Como já fora identificado que no primeiros vinte segundos de reprodução a taxa de transmissão do fluxo de vídeo alcança, aproximadamente, 28 Mbps por cliente, achava-se que não haveria comprometimento do canal. Porém, o resultado foi diferente. O cliente que iniciou a primeira reprodução permaneceu reproduzindo, mas o

segundo parou de reproduzir o filme após os 20". O experimento voltou a ser repetido sob as mesmas condições, porém sem apresentar as anomalias inicialmente descritas.

Aos 2'20" de reprodução, por cerca de aproximadamente cinco segundos, o Ultrabook teve a imagem ligeiramente alterada, sem, no entanto, perder o andamento do filme. Após 6'55", a transmissão do iPerf foi interrompida para aumentar a banda para 50 Mbps. E, aos 14' de reprodução, aumentou-se a carga para 60 Mbps. Em todas as situações, os pacotes gerados pelo iPerf foram limitados à taxa de 5 Mbps, sendo o restante descartado. As Figuras 5.11 e 5.12 representam o tráfego na interface do servidor de *streaming* e a carga gerada pelo iPerf, respectivamente.

Figura 5.11: Carga da interface de rede no servidor de *streaming*



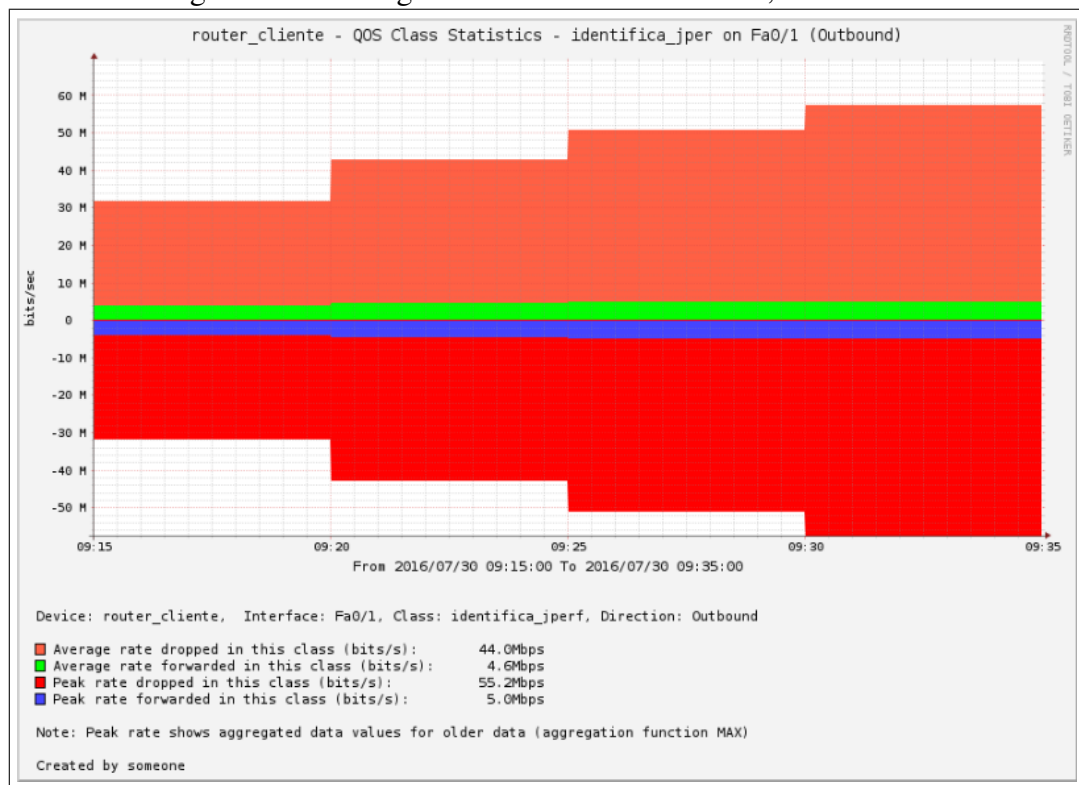
Fonte: *software* de monitoramento de redes Cacti

5.3 Cenário com roteadores Cisco nas instalações da UFS

Foram realizados cinco rodadas de 1 minuto e 2 segundos, seguindo a seguinte ordem de execução:

1. Com QoS aplicado nas interfaces de rede dos roteadores e sem carga do iPerf;

Figura 5.12: Tráfego iPerf no roteador Cisco, lado cliente



Fonte: *software* de monitoramento de redes Cacti

2. Com QoS aplicado nas interfaces de rede dos roteadores e com carga do iPerf.

Como não houve influência no cenário sem QoS em relação a aplicação do QoS nas interfaces de rede dos roteadores quando submetidas a testes sem carga do iPerf, optou-se por retirar as duas primeiras etapas dos testes realizados nas instalações do IFS.

5.3.1 Com QoS nas Interfaces de rede dos roteadores, com e sem carga do iPerf

A finalidade para se realizar os testes combinando, durante as 5 rodadas, aplicação ou não da carga do iPerf, se deu para evitar que o usuário criasse expectativas de fazer suas observações independentemente das condições de tráfego da rede. O procedimento seguiu as orientações descritas na seção 4.4 para a sua execução.

A tabela 5.2 representa a carga gerada pelo *software* iPerf entre um notebook na mesma rede dos dispositivos dos usuários e o servidor VLC.

Tabela 5.2: Carga gerada pelo iPerf e descartada pela política de QoS

Intervalo (sec)	Transfer (Kbytes/sec)	Largura de Banda (Kbits/sec)	Jitter (ms)	Perda/Total	Datagrams
0.0-12.3	7.125	4.764	15,491	89.984/94.947	(95%)
0.0-12.2	7.126	4.765	17,946	90.178/95.097	(95%)
0.0-12.3	7.129	4.767	14,515	90.125/95.031	(95%)
0.0-12.3	7.117	4.738	7,835	89.176/94.071	(95%)
0.0-12.2	7.127	4.766	18,36	90.035/94.976	(95%)
0.0-13.2	7.186	4.463	16,477	87.212/92.218	(95%)

Fonte: *software jPerf* versão para Windows

O campo Intervalo informa o tempo da carga submetida na rede em rodadas preestabelecidas, como será descrito nas subseções seguintes, na qual as 3 primeiras linhas referem-se ao tráfego injetado nos testes com o protocolo HTTP, e as 3 últimas linhas referentes aos testes com o protocolo RTSP; o campo *Transfer* diz respeito ao total de *bytes* por segundo transferidos como carga; a Largura de Banda representa a quantidade de *bits* que efetivamente foram encaminhados ao servidor VLC após passar pela política de QoS aplicada à interface de rede conectada a esse servidor; *jitter* aponta a variação causada na rede durante o intervalo de tempo da carga injetada; o campo Perda/Total informa do total transferido, quanto foi descartado entre cliente e servidor; por fim, *Datagrams* indica, em percentual, as perdas ocorridas durante a injeção do tráfego gerado pelo *iPerf*.

As experiências do usuário e dispositivo sofreram a influência do tráfego da tabela 5.2, que serviram de base para a avaliação dos dados classificados com o questionário aplicado durante as avaliações nas duas subseções a seguir.

Experiência do Usuário e do dispositivo – Protocolo HTTP

A experiência do usuário foi mensurada de acordo com a tabela 5.3, assim como a experiência do dispositivo da tabela 5.4, utilizando o protocolo HTTP.

Para essas condições, o tráfego injetado a partir do *iPerf*, ocorreu entre as rodadas 2, 3 e 4, durante 12 segundos, a partir do tempo de 15 segundos de execução do vídeo em R2, a partir de 30 segundos em R3 e, após 10 segundos do início, em R4.

Apenas 3 participantes relataram a experiência como EXCELENTE, mas somente 2 dispositivos obtiveram tal correlação (E3 e E6). Vale ressaltar que esses equipamentos não são os que possuem a maior velocidade de processamento, mas são os únicos que não são *smartphones*, que nos permite concluir que a arquitetura influencia no resultado da avaliação.

Tabela 5.3: Experiência do usuário – Protocolo HTTP

Qualidade do Vídeo	R1	R2	R3	R4	R5
Excelente (acima de 90%)			1	1	2
Boa (entre 70% e 89%)	1	2	1		1
Regular (entre 50% e 69%)			2	4	2
Ruim (entre 30% e 49%)	4	3	1	1	1
Impossível avaliar (abaixo de 30%)	1	1	1		

Fonte: Elaborado pelo autor com dados extraídos do *software* Cacti.

Tabela 5.4: Experiência do dispositivo – Protocolo HTTP

Qualidade do Vídeo	E1	E2	E3	E4	E5	E6
Excelente (acima de 90%)			1			3
Boa (entre 70% e 89%)		1	2	1		1
Regular (entre 50% e 69%)		3	1	1	3	
Ruim (entre 30% e 49%)	5		1	1	2	1
Impossível avaliar (abaixo de 30%)		1		2		

Fonte: Elaborado pelo autor com dados extraídos do *software* Cacti.

Essa constatação pode ser comprovada ao analisar o dispositivo E1 que teve desempenho ruim nas 5 rodadas, conforme tabela 5.3.

Os questionários foram elaborados para que o usuário relatasse, subjetivamente, sua experiência no caso de essa ter sido avaliada entre REGULAR e RUIM. Como exemplo, o usuário do dispositivo E1 relatou obter boa qualidade de áudio, porém impossibilitado para reprodução do vídeo.

Experiência do Usuário e do dispositivo – Protocolo RTSP

As tabelas 5.5 e 5.6 representam, respectivamente, as experiências dos usuários e de seus dispositivos.

Seguindo o mesmo princípio de injeção de tráfego com *iPerf* utilizado com o protocolo HTTP, foi injetado tráfego na rodada 2 a partir de 15 segundos do início da reprodução no servidor; carga adicionada após 11 segundos do início do vídeo e, após 30 segundos de reprodução.

Tabela 5.5: Experiência do usuário – Protocolo RTSP

Qualidade do Vídeo	R1	R2	R3	R4	R5
Excelente (acima de 90%)					
Boa (entre 70% e 89%)					2
Regular (entre 50% e 69%)	3		1	1	2
Ruim (entre 30% e 49%)	1	1	1	4	1
Impossível avaliar (abaixo de 30%)	2	5	4	1	1

Fonte: Elaborado pelo autor com dados extraídos do *software* Cacti.

Tabela 5.6: Experiência do dispositivo – Protocolo RTSP

Qualidade do Vídeo	E1	E2	E3	E4	E5	E6
Excelente (acima de 90%)						
Boa (entre 70% e 89%)		1		1		
Regular (entre 50% e 69%)			2	1		4
Ruim (entre 30% e 49%)	4	1	1	1	1	
Impossível avaliar (abaixo de 30%)	1	3	2	2	4	1

Fonte: Elaborado pelo autor com dados extraídos do *software* Cacti.

Nenhum usuário e, conseqüentemente, dispositivo, teve experiência qualificada como EXCELENTE; apenas 2 usuários entenderam que a sua experiência foi considerada como BOA.

Mesmo reduzindo a resolução do vídeo, esse protocolo não proporcionou bom desempenho aos dispositivos, mas aconteceu algo interessante: os dispositivos E3 e E6 que, outrora, tiveram seus respectivos usuários avaliando como EXCELENTE entre as rodadas R3 a R5 no protocolo HTTP, não foram capazes de avaliar sequer como BOA nesse protocolo. Os dispositivos E2 e E4, ou seja, 2 *smartphones*, obtiveram os melhores desempenhos nessa fase dos testes.

Os relatos dos usuários para suas experiências coincidem em quase a sua totalidade: muitas perdas de quadros, com conseqüentes "congelamentos" da imagem durante a reprodução, bem mais presente nesse protocolo do que a experiência com o HTTP.

5.4 Cenário com a placa NetFPGA

Inicialmente, o uso da placa NetFPGA foi pensado para estar conectado entre os dois roteadores Cisco. Mas, devido a problemas técnicos, optou-se por simular a experiência do usuário em dois cenários, cujas características desse são bem distintas do cenário com roteadores Cisco:

- o projeto desenvolvido para que a placa funcione como roteador não permite controle de QoS;
- as interfaces de rede da placa só permitem conexões a *Gigabit*.

Por um lado, foi interessante ter esse cenário com todas essas restrições porque, se não permite ajustes dos parâmetros de QoS da rede, também não aceita conexões diferentes de *Gigabit*. Ambas as restrições não são limitações da placa, mas de quem desenvolveu os projetos para uso com essa finalidade.

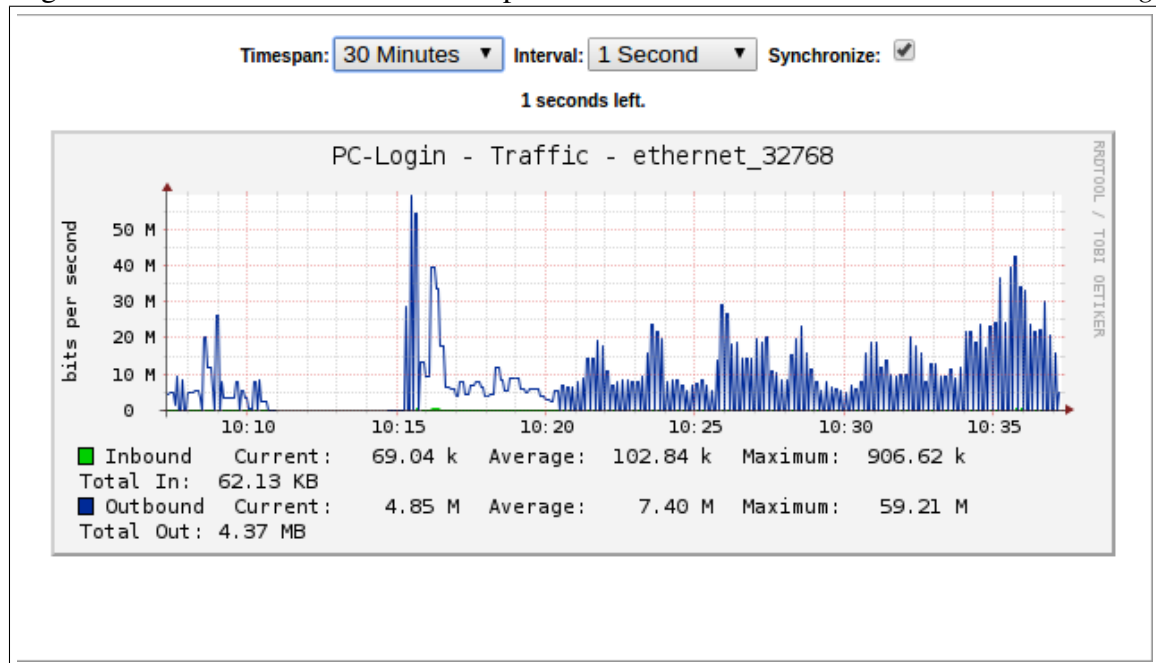
Nas subseções seguintes, foram extraídos os dados dos experimentos com e sem carga gerada pelo *iPerf*.

5.4.1 Sem QoS nas Interfaces da placa NetFPGA, sem carga do *iPerf*

O limite do canal se resume a taxa de transmissão do meio físico sem fio. O roteador utilizado tem todas as interfaces padrão RJ45 *Gigabit* – uma saída para a Internet e quatro saídas para LAN – e o padrão *wireless* é 802.11ac, podendo atingir até 450 Mbps, no caso do modelo usado nesse experimento. Como os testes com roteadores Cisco ocorreram com apenas duas estações, o mesmo cenário foi mantido. O resultado aconteceu como se esperava: nenhuma disputa devida a alta largura de banda disponível, ou seja, desde os primeiros segundos até o vigésimo segundo minuto sem quaisquer alterações na reprodução do filme. A Figura 5.13 demonstra a curva de transmissão dos fluxos do filme muito parecidas com as extraídas das interfaces de redes dos roteadores Cisco.

5.4.2 Sem QoS nas Interfaces da placa NetFPGA, com carga do *iPerf*

Como essa placa só aceita conexões *Gigabit*, e as interfaces *wireless* são padrão “n” e estavam conectadas a 300 Mbps (Figura 5.14), os testes com carga foram iniciados na taxa

Figura 5.13: Gráfico de transmissão a partir da interface de rede do servidor de *streaming*

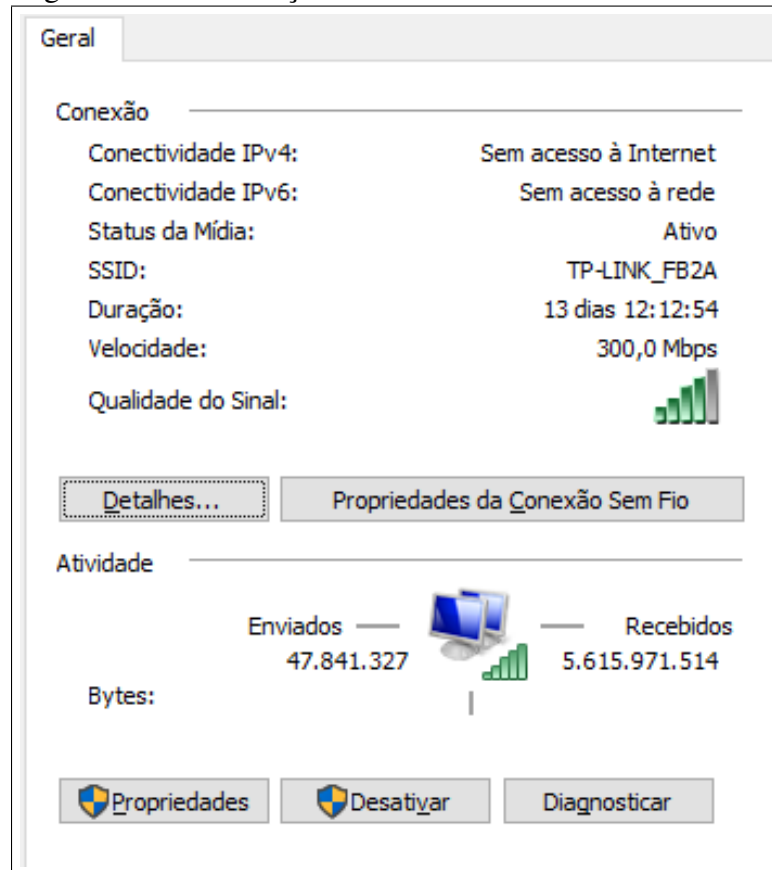
Fonte: *software* de monitoramento de redes Cacti

de 300 Mbps. Contudo, o Ultrabook parou de reproduzir após 47” de transmissão, e a taxa máxima ocupada pelo canal, de acordo com a Figura 5.15, foi de 280,89 Mbps.

5.5 Considerações finais do Capítulo

No ambiente de testes do IFS, um ponto a ser observado foi o filme como fator determinante nas condições da rede. Não importam regras de QoS se o comportamento de um fluxo exige que se garanta uma largura de banda mínima para que seja reproduzido. Todavia, ao deslocar o tempo de início em cada uma das estações, foi possível reproduzir o filme em até três dispositivos, ajustando a velocidade das interfaces que interliga os roteadores Cisco para 30 Mbps. A seguinte estratégia foi adotada: uma estação foi configurada com a opção “cache de rede” para armazenar 11” de *buffer*; a segunda, para com *cache* de 10”, e a terceira foi mantido o valor *default*, que é de um segundo. Após iniciar a transmissão, a ordem de reprodução foi do dispositivo com maior tempo de armazenamento, seguindo a lógica até o de menor valor. Como cada início foi diferenciado a priori, não houve concorrência do canal nas cenas, porque cada dispositivo estava reproduzindo em momentos distintos. Dessa

Figura 5.14: Informações da interface de rede do Ultrabook

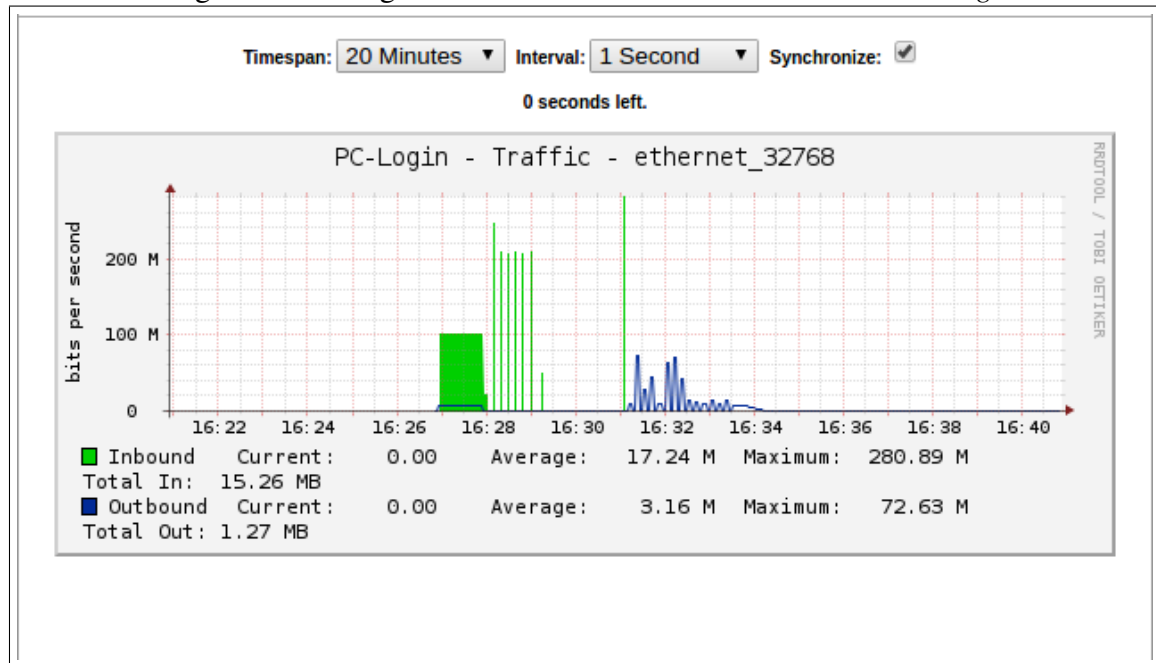


Fonte: sistema operacional Windows 10

forma, o filme pode ser reproduzido do começo ao fim sem paralisação, mas em momentos de maior atividade nas cenas, uma ou outra estação, momentaneamente, mudava a aparência da imagem, sem, no entanto, deixar de prosseguir com a reprodução.

O motivo de não manter o mesmo padrão de carga dos testes com roteadores Cisco se deu por razões óbvias: os enlaces possuíam capacidades diferentes, com enorme vantagem para o segundo cenário. Mas, as leis da físicas também se aplicaram nesse teste, e a solução para o problema seguiu a mesma lógica imposta no cenário com roteadores Cisco, ou seja, adicionar *cache* de rede em uma das estações para que as cenas com maior rajada de fluxos não se coincidissem. O resultado final foi uma reprodução limpa, do início ao final do tempo estabelecido.

Quanto aos ensaios no ambiente da UFS, observou-se que a injeção de tráfego via *iPerf* pouco influenciou a experiência do usuário. O protocolo HTTP proporcionou melhor ex-

Figura 5.15: Carga na interface de rede do servidor de *streaming*.

Fonte: *software* de monitoramento de redes Cacti

periência em relação ao RTSP. As políticas de QoS foram eficazes para garantir que tráfegos diferentes dos fluxos multimídia não sofressem alteração em seu percurso. Os dispositivos com menor capacidade de processamento não proporcionaram boa experiência para o usuário, mas houveram dispositivos que, mesmo contendo boas características em sua arquitetura computacional, sofreram algum tipo de interferência.

Há outros fatores a serem observados, tais como *software* de geração/reprodução de *streaming* multimídia e equipamento para servir como infraestrutura para redes sem fio.

Capítulo 6

Considerações Finais

A Experiência do Usuário, apesar da sua subjetividade na avaliação, pode ser gerenciada através dos serviços da pilha de protocolos do modelo de referência TCP/IP. Não há controle sobre as variáveis que se apresentam durante uma transmissão qualquer de fluxos multimídia. Nos testes realizados, buscou-se esgotar os recursos de rede na tentativa de fazer com que os parâmetros de QoS pudessem modelar tráfegos indesejáveis. Como exemplo, foi aplicado um tráfego simulado, via *iPerf*, ocupando 100 Mbps. Mesmo tendo colocado limites na admissão desse tipo de fluxo, por alguma razão que não foi possível explicar, a regra não descartava todo o excedente, concorrendo com a banda não totalmente ocupada pelo *streaming* de vídeo. Apesar dos manuais da Cisco explicar claramente as funções dos algoritmos implementados em seus roteadores, e ela fazer parte da padronização do padrão DSCP, a “inteligência”, ou seja, a forma como essas políticas de controle funcionam, são totalmente omitidas para a comunidade em geral. Mas é aqui que essa limitação motiva a comunidade científica a investir em pesquisas sobre qualidade de serviços.

Outro fator de grande impacto nessa experiência é realmente o dispositivo do usuário final. A percepção sobre o que se esperava de resultado *versus* o obtido foi muito próximo, e a velocidade de processamento é o que determina se filmes em FULL HD poderão ou não serem reproduzidos. Notebook com aceleradora gráfica, tendo 256 MB de RAM dedicada e 6 GB de RAM não foi capaz de proporcionar boa experiência, porque o processador, apesar de *dual core* e disco rígido SSD (*Solid State Drive*), seu *clock* era de apenas 1 Ghz.

E um terceiro fator, mas não menos importante, é a qualidade do sinal *wifi*. No único experimento que envolveu os usuários em pontos devidamente posicionados, haviam muitas

pessoas com seus dispositivos móveis, e a concorrência com outros pontos de acesso presentes. A consequência foi um desempenho abaixo do esperado, mesmo tendo dispositivo com ótimas características de processamento. Quando o ambiente ficou vazio, ao repetir o experimento, o resultado da reprodução teve desempenho satisfatório, com as limitações físicas dos dispositivos participantes apenas como causadores da boa ou má experiência para o usuário.

Dessa maneira, a infraestrutura (espaço físico e acesso à rede) também é fator determinante para que o usuário possa, ou não, ter uma boa experiência durante a reprodução de um *streaming* de vídeo.

6.1 Trabalhos Futuros

A diversidade de situações adversas forçaram replanejamentos nos experimentos, proporcionando alguns possíveis trabalhos futuros para maiores investigações:

- Estudo aprofundado do protocolo UDP – apesar de muito citado na literatura como o protocolo para uso em aplicações que não exijam controle de fluxo, tal como fluxos multimídia, este se mostrou com baixa qualidade mesmo em um enlace ponto a ponto, cujos equipamentos estavam conectados a um cabo Cat.6 sem elemento ativo entre eles;
- Estudo aprofundado do *codec* MPEG4, uma vez que os quadros I se perderam com certa frequência quando a aplicação utilizava o protocolo RTSP sobre UDP. Apesar de, aparentemente, o *software* VLC ser executado de maneira esperada, também pode ser um dos causadores de tal comportamento. O objetivo desse trabalho foi avaliar a experiência do usuário considerando os aspectos da rede, características da aplicação e de dispositivos finais, mas não identificar as causas individuais de cada um dos elementos citados;
- Estudo aprofundado sobre os algoritmos de QoS que tratam das filas e *buffers* de rede;
- Desenvolver projetos na placa NetFPGA que dê suporte a análises dos parâmetros de QoS;
- Identificar/implementar formas de garantir que o serviço de *streaming* possa alocar a sessão do cliente conforme a demanda.

Essa pesquisa proporcionou um mergulho um pouco mais profundo no universo das redes de computadores. Ao fazer esse mergulho, pode-se perceber o muito que ainda precisa ser

melhorado. As portas já foram abertas através de outras pesquisas do passado, que serviram de base para essa, mas que deu a certeza de que não deve parar por aqui.

REFERÊNCIAS

AGENCY, D. A. R. P. *RFC791 - Internet Protocol*. 1981. 45 p. Disponível em: <http://www.ietf.org/rfc/rfc791>.

ALCATEL-LUCENT. *QoS Policies*. 2012. Disponível em: <https://infoproducts.alcatel-lucent.com/>.

ALRESHOODI, M.; WOODS, J. Survey on QoE\QoS Correlation Models for Multimedia Services. *International Journal of Distributed and Parallel systems*, 2013. ISSN 22293957.

AROUSSE, S.; MELLOUK, A. Survey on machine learning-based QoE-QoS correlation models. *2014 International Conference on Computing, Management and Telecommunications, ComManTel 2014*, p. 200–204, 2014.

BLAKE, S. et al. *RFC 2475 - An Architecture for Differentiated Services*. 1998. 1–36 p. Disponível em: <https://www.rfc-editor.org/rfc/pdf/rfc2475.txt.pdf>.

BOTIA, D. J. *An approach to correlation of QoE/QoS metrics over IPTV using a Diffserv Network*. 2012. 6 p. Disponível em: <https://www.researchgate.net/>.

BRADNER, S.; UNIVERSITY, H. *No Title*. 1997. 1–3 p. Disponível em: <https://www.rfc-editor.org/rfc/pdf/rfc2119.txt.pdf>.

BROOKS, P.; HESTNES, B. r. User measures of quality of experience: Why being objective and quantitative is important. *IEEE Network*, v. 24, n. April, p. 8–13, 2010. ISSN 08908044.

CETIC.BR. *VELOCIDADE MÁXIMA PARA DOWNLOAD CONTRATUALMENTE FORNECIDA PELO PROVEDOR DE INTERNET NOS ÚLTIMOS 12 MESES*. 2014. s p. Disponível em: <http://www.cetic.br/tics/osfil/2014/geral/A10/>.

CISCO. *QoS: Regulating Packet Flow Configuration Guide, Cisco IOS Release 12.4*. 2011. 36 p. Disponível em: <http://www.cisco.com/>.

Cisco Systems, I. *Comparing the bandwidth and priority Commands of a QoS Service Policy*. 2008. 9 p.

DAVIE, B. et al. *RFC 3246 - An Expedited Forwarding PHB*. 2002. 1–16 p. Disponível em: <https://tools.ietf.org/pdf/rfc3246.pdf>.

DVORAK, C. *Definition of Quality of Experience (QoE)*. Genebra, 2007. 2 p. Disponível em: <https://www.itu.int/>.

- FELD, T. H. et al. Quantification of YouTube QoE via crowdsourcing. *Proceedings - 2011 IEEE International Symposium on Multimedia, ISM 2011*, p. 494–499, 2011.
- FIEDLER, M.; HOSSFELD, T.; TRAN-GIA, P. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, v. 24, n. April, p. 36–41, 2010. ISSN 08908044. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={\& }arnumber=5430>.
- HAN, B. et al. QoE model based optimization for streaming media service considering equipment and environment factors. *Wireless Personal Communications*, v. 66, p. 595–612, 2012. ISSN 09296212.
- HEINANEN, J. et al. *RFC 2597 - Assured Forwarding PHB Group*. 1999. 1–11 p. Disponível em: <https://www.rfc-editor.org/rfc/pdf/rfc2597.txt.pdf>.
- HERSENT, O.; GURLE, D.; PETIT, J.-P. Telefonía IP: Comunicação multimídia baseada em pacotes. In: . São Paulo: Prentice Hall, 2002. p. 451. ISBN 85-88639-02-5.
- HOHLFELD, O. et al. A QoE Perspective on Sizing Network Buffers. *Proceedings of the 2014 Conference on Internet Measurement Conference - IMC '14*, ACM Press, New York, New York, USA, p. 333–346, 2014. Disponível em: <http://dl.acm.org/citation.cfm?doid=2663716.2663730>.
- HSU, W.-H. et al. A qoe-based measurement for diffserv multicasting networks. In: *Computational Science and Its Applications (ICCSA), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 183–190.
- IBOPE. *Brasil é o terceiro país em número de usuários ativos na internet*. 2013. Disponível em: <http://www.ibope.com.br/pt-br/noticias/paginas/brasil-e-o-terceiro-pais-em-numero-de-usuarios-ativos-na-internet.aspx>.
- JACAUNA, R.; MORENO, E. D.; SALGUEIRO, R. J. P. D. B. User Experience Evaluation Based on Arbitration QoS Parameters in Video Stream Using NetFPGA in a Controlled Environment. In: . [S.l.: s.n.], 2016. p. 20–23. ISBN 9781612084732.
- KATSARAKIS, M.; THEODOSIADIS, V.; PAPADOPOULI, M. *On the Evaluation of a User-centric QoE-based Recommendation Tool for Wireless Access*. [S.l.], 2014. Disponível em: <http://www.ics.forth.gr>.
- MEHMOOD, M. et al. Qoe-lab: Towards evaluating quality of experience for future internet conditions. In: KORAKIS, T. et al. (Ed.). *Testbeds and Research Infrastructure. Development of Networks and Communities*. Springer Berlin Heidelberg, 2012, (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, v. 90). p. 286–301. ISBN 978-3-642-29272-9. Disponível em: http://dx.doi.org/10.1007/978-3-642-29273-6_22.
- NETFPGA.ORG. *NetFPGA*. 2014. s p. Disponível em: <http://netfpga.org/2014/>.
- NETFPGA.ORG. *NetFPGA-CML*. 2014. s p. Disponível em: <http://netfpga.org/2014/#/systems/2netfpga-1g-cml/details/>.

- OIESTAD, R. *VideoLAN - VLC media player for Fedora*. 2015. Disponível em: <http://www.videolan.org/vlc/download-fedora.html>.
- ORDONEZ, E. D. M. *Novos Algoritmos de Compressao de Codigo de Sistemas Embarcados*. Aracaju/SE: PROJETO DO EDITAL UNIVERSAL do CNPq, 2014. 15 p.
- OROSZ, P. et al. A case study on correlating video QoS and QoE. *2014 IEEE Network Operations and Management Symposium (NOMS)*, Ieee, p. 1–5, maio 2014. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6838399>.
- PINSON, M.; WOLF, S. A new standardized method for objectively measuring video quality. *Broadcasting, IEEE Transactions on*, v. 50, n. 3, p. 312–322, Sept 2004. ISSN 0018-9316.
- RIVERA, S. et al. QoS-QoE Correlation Neural Network Modeling for Mobile Internet Services. *2013 International Conference on Computing Management and Telecommunications Commantel*, p. 75–80, 2013.
- SALGUEIRO, R. J. P. d. B. *Uma Metodologia para Caracterização de Tráfego de Vídeo baseada nos momentos de Primeira e Segunda Ordem*. 176 p. Tese (Doutorado) — Universidade Federal de Pernambuco, 2004.
- SILLER, M.; WOODS, J. Improving quality of experience for multimedia services by QoS arbitration on a QoE framework. in *Proc. of the 13th Packed Video Workshop 2003*, 2003. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.194.7031&rep=rep1&type=pdf>.
- SYSTEMS, C. *Standards Supported in Cisco IOS and IOS-XE Software*. 2011. 1–25 p. Disponível em: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps8802/ps6968/ps6350/prod{__}bulletin0900aecd802eaa4f.h.
- TANENBAUM, A. S.; WETHERALL, D. J. *Redes de Computadores*. 5a. ed. São Paulo: Pearson Education do Brasil, 2011. 583 p. ISBN 978.85-7605-924-0.
- VARELA, M.; LAULAJAINEN, J. Qoe-driven mobility management; integrating the users' quality perception into network; level decision making. In: *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*. [S.l.: s.n.], 2011. p. 19–24.
- XILINX, C. *7 Series FPGAs Packaging and Pinout Product Specification User Guide (UG475).pdf*. 2014. 288 p.

Apêndice A

Configuração dos roteadores e Questionário

As configurações abaixo, parte delas, se aplicam a ambos, outras partes (as interfaces de rede) mudarão as políticas de QoS em relação ao sentido dos fluxos.

Roteador conectado ao servidor de *streaming*

Classificação dos pacotes

```
class-map match-all identifica_jperf
  match ip dscp cs1
  !
class-map match-all identifica_streaming
  match ip dscp ef
  !
class-map match-any marca_jperf
  match access-group name jperf
  !
class-map match-any marca_streaming
  description marca os pacotes para fluxos multimedia
  match access-group name server_video
```

Marcação do pacotes

```
policy-map marca_streaming
  class marca_streaming
  set ip dscp ef
  !
policy-map limita_jperf
  class identifica_jperf
  shape average percent 5
  police cir percent 5
  bandwidth percent 5
  service-policy marca_streaming
  !
policy-map prioriza_streaming
  class identifica_streaming
  bandwidth percent 30
  police cir percent 30
  !
policy-map marca_jperf
  class marca_jperf
  set ip dscp cs1
  !
```

Configuração das interfaces

```
interface FastEthernet0/0
  description LAN connection to DLINK.ROUTER
  ip address 192.168.0.1 255.255.255.252
  speed 100
  full-duplex
  service-policy input marca_jperf
  service-policy output prioriza_streaming
```

```
!  
interface FastEthernet0/1  
description LAN connection to ROUTER_VIDEO  
ip address 172.31.0.2 255.255.255.252  
bandwidth 30000  
speed auto  
full-duplex  
service-policy input marca_streaming  
service-policy output limita_jperf  
!
```

Rotas estáticas

```
ip route 10.0.0.0 255.255.255.0 FastEthernet0/0  
ip route 192.168.10.0 255.255.255.0 FastEthernet0/1
```

Lista de Controle de Acesso – ACL

```
ip access-list extended jperf  
permit udp any any eq 5001  
ip access-list extended server_video  
permit ip host 192.168.10.10 any  
!
```

Configuração dos *traps* SNMP

```
snmp-server community netfpga RO  
snmp-server ifindex persist  
snmp-server ip dscp 10  
snmp-server enable traps snmp authentication linkdown  
linkup coldstart warmstart  
!
```

Configuração de sincronização do relógio

```
ntp server 192.168.10.10
```

Figura A.1: Formulário para avaliar a experiência do usuário.

**Questionário para mensurar a Qualidade de Experiência do vídeo
"INSURGENTE".**

Objetivo: quantificar a experiência do usuário de serviços de streaming de vídeo a partir da sua percepção.

Qualidade do vídeo no tempo entre 00:00 a 01:02

Avalie a qualidade do vídeo conforme os parâmetros a seguir.

	Rodadas				
	R1	R2	R3	R4	R5
Excelente (acima de 90%)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Boa (entre 70% e 89%)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regular (entre 50% e 69%)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ruim (entre 30% e 49%)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Impossível avaliar (abaixo de 30%)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

OBS: caso julgue a sua experiência como REGULAR ou abaixo disso, favor descrever o MOTIVO.

Rodada 1: _____

Rodada 2: _____

Rodada 3: _____

Rodada 4: _____

Rodada 5: _____

Equipamento: _____